



The Eleventh Conference on Language Engineering

**December 14-15, 2011, Cairo, Egypt
(ESOLEC'2011)**

Organized by

Egyptian Society of Language Engineering (ESOLE)

Under the Auspices of

**PROF. DR. MOHAMED ELTOKHY
Dean, Faculty of Engineering, Ain Shams University**

**CONFERENCE CHAIRPERSON
PROF. DR. M. A. R. GHONAIMY**

**CONFERENCE COCHAIRPERSON
PROF. DR. SALWA ELRAMLY**

**Faculty of Engineering –Ain Shams University
Cairo, Egypt**

<http://esole-eg.org>

Conference Chairman:

Prof. Dr. M. R. A. Ghonaimy

Technical Program Committee:

Prof. Taghrid Anber , **Egypt**
Prof. I. Abdel Ghaffar , **Egypt**
Prof. M. Ghaly, **Egypt**
Prof. M. Z. Abdel Mageed, **Egypt**
Prof. Khalid Choukri, ELDA, **France**
Prof. Nadia Hegazy, **Egypt**
Prof. Christopher Ciri, LDC, **U.S.A**
Prof. Mona T. Diab, Stanford U., **U.S.A**
Prof. Ayman ElDossouki, **Egypt**
Prof. Afaf AbdelFattah, **Egypt**
Prof. Y. ElGamal, **Egypt**
Prof. M. Elhamalaway, **Egypt**
Prof. S. Elramly, **Egypt**
Prof. H. Elshishiny, **Egypt**
Prof. A. A. Fahmy, **Egypt**
Prof. I. Farag, **Egypt**
Prof. Magdi Fikry, **Egypt**
Prof. Wafa Kamel, **Egypt**
Prof. S. Krauwer, **Netherlands**
Prof. Bente Maegaard, CST, **Denmark**
Prof. A. H. Moussa, **Egypt**
Prof. M. Nagy, **Egypt**
Prof. A. Rafea, **Egypt**
Prof. Mohsen Rashwan, **Egypt**
Prof. H.I. Shaheen, **Egypt**
Prof. S.I. Shaheen, **Egypt**
Prof. Hassanin M. AL-Barhamtoshy, **Egypt**
Prof. M. F. Tolba, **Egypt**
Dr. Tarik F. Himdi, **Saudi Arabia**

Organizing Committee

Prof. I. Farag	Prof. S. Elramly
Prof. Taghride Anbar	Prof. Hany Kamal
Prof. H. Shahein	Dr. Mostafa Aref
Dr. Passant El-kafrawy	Dr. Fatma Newaigy
Eng. A Mausad	Eng. Manar Ahmed
Eng. Mona Zakaria	Eng. Bassant A. Hamid

Conference Secretary General

Prof. Dr. Salwa Elramly

Conference Sponsors



Scope of the Conference :

- **Language analysis and comprehension**
- **Language generation**
- **Spoken language understanding**
- **Discourse & dialogue systems**
- **Evaluation of natural language processing systems**
- **Large corpora**
- **Speech processing recognition and synthesis**
- **Natural language processing for information retrieval**
- **Machine translation**
- **Language engineering frameworks & methodologies**
- **Language engineering & artificial intelligence**
- **Automatic character recognition**
- **Semantic Web and Ontology Languages**
- **Mobile Web**
- **Social networks and contents development challenges**

*The Eleventh Conference on Language Engineering
Final Program*

Wednesday 14 December 2011

9.00 - 10.00 Registration

10.00 - 10.30 Opening Session

10.30 - 11.30 **Session 1: Invited Paper 1:**

Chairman: Prof. Dr. M. Adeeb Riad Ghonaimy

From Data to Nuanced Information: Making Implicit Knowledge Useful

Mona Diab

Columbia University, USA

11.30 - 12.15 Coffee break

12.15 - 14.00 **Session 2: Natural Language Processing for Information Retrieval**

Chairman: Prof. Dr. Ibrahim Farag

1. Analyzing Arabic Diacritization Errors of MADA and Sakhr Diacritizer

Hamdy Mubarak, Ahmed Metwally, Mostafa Ramadan

Arabic NLP Researches, Sakhr Software

2. SAFAR platform and its morphological layer

Younes Souteh and Karim Bouzoubaa

Mohammadia School of Engineers, Mohammed 7th University - Adgal, Rabat, Morocco.

3. Arabic Information Retrieval: How to Get “Good” Results at a Lower Cost?

Claude Audebert*, André Jaccarini*, Christian Gaubert**

**Maison méditerranéenne des sciences de l’homme (MMSH)*

***Institut français d’archéologie orientale du Caire (IFAO)*

4. Representing Arabic Documents Using Controlled Vocabulary Extracted from Wikipedia

Mohamed I. Eldesouki*, Waleed M. Arafa*, Kareem Darwish**, Mervat H. Gheith*

**Department of Computer and Information Sciences, Institute of Statistical Studies and Research, Cairo University, Egypt*

***Qatar Computing Research Institute, Qatar Foundation, Qatar*

14.00 - 15.00 Lunch

15.00 - 17.00 **Session 3: Machine Translation:**

Chairman: Prof. Dr. Mohamad Zaki Abdel Mageed

1. Linguistically Motivated Reordering Constraints for Phrase-based SMT: Base Phrase Chunks and Predicate Argument Structures

Mahmoud Ghoneim*, Marine Carpuat**, Mona Diab*

**Center for Computational Learning Systems, Columbia University, USA*

***NRC Institute for Information Technology, CANADA*

2. English to Arabic Statistical Machine Translation System Improvements using Preprocessing and Arabic Morphology Analysis

Shady Abdel Ghaffar*, Mohamed Waleed Fakhri**

**Faculty of computing and Information Technology, Arab Academy for Science and Technology, Sheraton, Cairo, Egypt*

***Faculty of Engineering, Electrical Engineering Department, University of Bahrain, Eissa Town, Bahrain*

3. Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas

Sameh Alansary

Phonetics and Linguistics Department Faculty of Arts, Alexandria University, Alexandria, Egypt.

4. The UNL Editor: A Manual Tool for Semantic Annotation

Sameh Alansary*, Magdy Nagi**, Noha Adly**

**Phonetic and Linguistics Department, Faculty of Arts, University of Alexandria ElShatby, Alexandria, Egypt*

***Computer and System Engineering Department, Faculty of Engineering Alexandria University, Egypt*

Thursday 15 December 2011

10.00 - 11.00 **Session 4: Room A: Language Engineering and Artificial Intelligence**

Chairman: Prof. Dr. Taghride Anbar

1. Mining Opinion in Arabic Data: A Comparison Between Supervised and Unsupervised Classification Approaches

Ahmed M. Misbah and Ibrahim F. Imam

Computer Science Department, Faculty of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Egypt.

2. Generating Lexical Resources for Opinion Mining in Arabic Language Automatically

Hanaa Bayomi Ali *, Mohsen Rashwan **, Samir Abd_Elrahman*

*Computer Science Department, Faculty of Computers and Information, Cairo University, Egypt.

** Electronics and Communications Department, Faculty of Engineering, Cairo University, Egypt.

3. Tapping into the Power of Automatic Scoring

Wael H. Gomaa, Aly A. Fahmy

Computer Science Department, Faculty of Computers & Information, Cairo University, Egypt.

11.00 - 11.45 **Session 5: Invited Paper 2:**

Chairman: Prof. Dr. Ibrahim Farag

اللسانيات الحاسوبية من منظور مجتمع المعرفة

د. نبيل على

خبير اللسانيات الحاسوبية

11.45 - 12.15 Coffee Break

12.15 - 13.00 **Session 6: Room A: Invited paper 3:**

Chairman : Prof. Dr. Mohsen Rashwan

أثر تجاوز صوتي الفعل الثلاثي المضعف في بابہ الصرفي:
دراسة لغوية حاسوبية على الأصوات الذلقية (ر-ل-ن)

أ.د/ وفاء كامل فايد

كلية الآداب – جامعة القاهرة

13.00 - 14.30 **Session 7: Room A: Language Analysis and Comprehension:**

Chairman : Prof. Dr. Hani Mahdi

1. Persian Morphology: Description and Implementation

Vahid R. Mirzaeian

ELT Department, Faculty of Engineering, Iran University of Science and Technology, Farjam Street, Narmak, Tehran, Iran

2. Correctness, Strength and Similarity Evaluation of Stemming Algorithms for Arabic

Daoud Daoud*, Christian Boitet**

*Princess Sumaya University for Technology

**GETALP, LIG, Université Joseph Fourier

3. Comparative Corpus-Based Study of the Complement Structure of the Verb “Said” and “Qala” in English and Arabic

Ateka Nasher^{*}, Sameh Al-Ansary^{**}, and Shadia El-Soussi^{***}

^{}English Language and Literature Department, Linguistics Branch, Faculty of Arts, Alexandria University, Alexandria, Egypt.*

*^{**}Phonetics and Linguistics Department, Faculty of Arts, Alexandria Univer University, Alexandria, Egypt.*

*^{***}Institute of Applied Linguistics, Faculty of Arts, Alexandria University, Alexandria, Egypt*

4. Automatic Speech Annotation Using HMM based on Best Tree Encoding (BTE) Feature

Amr M. Gody, Rania Ahmed Abul Seoud , Mohamed Hassan
Electrical Engineering Department, Fayoum University, Egypt

14.30 - 15.30 Lunch

15.30 - 16.30 **Session 8: Room A: Semantic Web and Ontology Languages:**

Chairman : Prof. Dr. Hassanin El-Barhamtoushy

1. An Enhanced Method for Ranking Arabic Web Pages Using Morphological Analysis

Esraa Abd Elraouf , Nagwa Lotfy Badr, Mohamed Fahmy Tolba
Faculty of Computer Science and Information Sciences, Ain Shams University, Cairo, Egypt

2. Text Generation Model from Rich Semantic Representations

Dalia Sayed, Mostafa Aref, Ibrahim Fathy
Department of Computer science, Faculty of Computer and Information Sciences, Ain-Shams University, Cairo, Egypt.

3. Implementation of Establishing Global Ontology by Matching and Merging

Susan Faisal Ellakwah^{*}, Passent El-Kafrawy^{**}, Mohamed Amin^{**},
El-Sayed El-Azhary^{*}

^{}Central Lab for Agricultural Expert Systems (CLAES), Agricultural Research Center (ARC), Giza, Egypt*

*^{**}Mathematics and CS Department, Faculty of Science, Menoufia University, Egypt*

16.30 - 17.00 **Session 9: Room A: Closing Session**

Chairman: Prof. Dr. Mohamed Younis Elhamalawy

Table of Contents

Page

I. Language Engineering Frameworks and Methodologies :

1. **Invited paper (1): From Data to Nuanced Information: Making Implicit Knowledge Useful** 1
Dr. Mona Diab
Columbia University, USA

II. Language Analysis and Comprehension

2. **Persian Morphology: Description and Implementation** 2
Vahid R. Mirzaeian
ELT Department, Faculty of Engineering, Iran University of Science and Technology, Farjam Street, Narmak, Tehran, Iran
3. **Correctness, Strength and Similarity Evaluation of Stemming Algorithms for Arabic** 10
Daoud Daoud*, Christian Boitet**
**Princess Sumaya University for Technology, Jordan*
***GETALP, LIG, Université Joseph Fourier, France*
4. **A Comparative Corpus-Based Study of the Complement Structure of the Verb “Said” and “Qala” in English and Arabic** 17
Ateka Nasher*, Sameh Al-Ansary**, and Shadia El-Soussi***
**English Language and Literature Department, Linguistics Branch, Faculty of Arts, Alexandria University, Alexandria, Egypt.*
***Phonetics and Linguistics Department, Faculty of Arts, Alexandria Univer University, Alexandria, Egypt.*
****Institute of Applied Linguistics, Faculty of Arts, Alexandria University, Alexandria, Egypt*

III. Language Engineering and Artificial Intelligence

5. **Mining Opinion in Arabic Data: A Comparison between Supervised and Unsupervised Classification Approaches** 25
Ahmed M. Misbah and Ibrahim F. Imam
Computer Science Department, Faculty of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport

6. **Generating Lexical Resources for Opinion Mining in Arabic Language Automatically** 35
 Hanaa Bayomi Ali *, Mohsen Rashwan **, Samir Abd_Elrahman*
 *Computer Science Department, Faculty of Computers and Information, Cairo University
 ** Electronics and Communications Department, Faculty of Engineering, Cairo University, Egypt.
7. **Tapping into the Power of Automatic Scoring** 42
 Wael H. Gomaa, Aly A. Fahmy
 Computer Science Department , Faculty of Computers & Information, Cairo University, Egypt
- IV. Semantic Web and Ontology Languages:**
8. **An Enhanced Method for Ranking Arabic Web Pages Using Morphological Analysis** 49
 Esraa Abd Elraouf , Nagwa Lotfy Badr, Mohamed Fahmy Tolba
 Faculty of Computer Science and Information Sciences, Ain Shams University, Cairo, Egypt
9. **Text Generation Model from Rich Semantic Representations** 58
 Dalia Sayed, Mostafa Aref, Ibrahim Fathy
 Department of Computer science, Faculty of Computer and Information Sciences,
 Ain-Shams University, Cairo, Egypt.
10. **Implementation of Establishing Global Ontology by Matching and Merging** 68
 Susan Faisal Ellakwah*, Passent El-Kafrawy**, Mohamed Amin**, El-Sayed El-Azhary*
 *Central Lab for Agricultural Expert Systems (CLAES), Agricultural Research Center (ARC), Giza, Egypt
 **Mathematics and CS Department, Faculty of Science, Menoufia University, Egypt
- V. Natural Language Processing for Information Retrieval**
11. **Analyzing Arabic Diacritization Errors of MADA and Sakhr Diacritizer** 82
 Hamdy Mubarak, Ahmed Metwally, Mostafa Ramadan
 Arabic NLP Researches, Sakhr Software

12. **SAFAR Platform and its Morphological Layer** 96
 Younes Souteh and Karim Bouzoubaa
*Mohammadia School of Engineers, Mohammed Vth University
 - Adgal, Rabat, Morocco.*
13. **Arabic Information Retrieval: How to Get “Good” Results at a Lower Cost?** 104
 Claude Audebert*, André Jaccarini*, Christian Gaubert**
 **Maison méditerranéenne des sciences de l’homme (MMSH)*
 ***Institut français d’archéologie orientale du Caire (IFAO)*
14. **Representing Arabic Documents Using Controlled Vocabulary Extracted from Wikipedia** 109
 Mohamed I. Eldesouki*, Waleed M. Arafa*, Kareem Darwish**, Mervat H. Gheith*
 **Department of Computer and Information Sciences, Institute of Statistical Studies and Research, Cairo University*
 ***Qatar Computing Research Institute, Qatar Foundation*
- VI. Machine Translation**
15. **Linguistically Motivated Reordering Constraints for Phrase-based SMT: Base Phrase Chunks and Predicate Argument Structures** 115
 Mahmoud Ghoneim*, Marine Carpuat**, Mona Diab*
 **Center for Computational Learning Systems, Columbia University, USA*
 ***NRC Institute for Information Technology, CANADA*
16. **English to Arabic Statistical Machine Translation Employing Pre-processing and Morphology Analysis** 122
 Shady Abdel Ghaffar, Mohamed Waleed Fakhre
 **Faculty of computing and Information Technology, Arab Academy for Science and Technology, Sheraton, Cairo, Egypt*
 ***Faculty of Engineering, Electrical Engineering Department, University of Bahrain Eissa Town, Bahrain*
17. **Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas** 128
 Sameh Alansary
*Bibliotheca Alexandrina, Alexandria, Egypt.
 Faculty of Arts, Alexandria University Alexandria, Egypt.*

18. **UNL Editor: An Annotation Tool for Semantic Analysis** 138
Sameh Alansary*, Magdy Nagi**, Noha Adly**
**Phonetic and Linguistics Department, Faculty of Arts, University
of Alexandria
ElShatby, Alexandria, Egypt
**Computer and System Engineering Department, Faculty of
Engineering Alexandria University, Egypt*

VII. Speech Processing

19. **Automatic Speech Annotation Using HMM based on Best Tree
Encoding (BTE) Feature** 153
Amr M. Gody, Rania Ahmed Abul Seoud , Mohamed Hassan
Electrical Engineering Department, Fayoum University

VIII. Computational Linguistics

20. أثر تجاوز صوتي الفعل الثلاثي المضعف في بابهِ الصرفي:
دراسة لغوية حاسوبية على الأصوات الذلّقية (ر-ل-ن)
أ.د/ وفاء كامل فايد
كلية الآداب – جامعة القاهرة 160
21. اللسانيات الحاسوبية من منظور مجتمع المعرفة
د. نبيل على
خبير اللسانيات الحاسوبية 188

Linguistically Motivated Reordering Constraints for Phrase-based SMT: Base Phrase Chunks and Predicate Argument Structures

Mahmoud Ghoneim^{*1}, Marine Carpuat^{**2}, Mona Diab^{*3}

**Center for Computational Learning Systems, Columbia University
475 Riverside Drive, New York, NY 10115, USA*

¹mghoneim@ccls.columbia.edu

³mdiab@ccls.columbia.edu

***NRC Institute for Information Technology
283 Alexandre-Taché, Gatineau, Quebec J8X 3X7, Canada*

²Marine.Carpuat@cnrc-nrc.gc.ca

Abstract: We compare the impact of conventional distance-based reordering constraints in phrase-based statistical machine translation (SMT) with two new reordering constraints that rely on boundaries defined by linguistic preprocessing of the SMT input: (1) base-phrase chunking and (2) argument boundary detection from semantic role labeling. While the different constraints yield very close scores with automatic metrics of translation quality, manual analysis of translations show that each constraint has different strengths and weaknesses.

1 INTRODUCTION

Phrase-based Statistical Machine Translation (SMT) models have improved translation quality by focusing on learning large phrasal translation lexicons. While phrasal translations capture very local reorderings, longer range structural differences between input and output languages are not explicitly modeled. Reordering is simply made possible by translating input phrases out of their original order.

Unlike in syntax-based SMT, where the reordering of input phrases is driven by syntactically-motivated rules, most phrase-based statistical machine translation systems use weak reordering models, penalizing reorderings based on distance only and relying on the output language model to evaluate the well-formedness and fluency of the output sentence. These models typically allow local reorderings and prevent long-distance reordering, thus implicitly capturing the intuition that neighboring words in the input tend to be related words and should be translated as neighboring words. The reordering model has no knowledge of the syntactic or semantic constituents that should be preserved in translation.

In this paper, we investigate whether the maximum distance reordering constraint is a good approximation for capturing meaningful subsentential units, and compare and contrast its impact with that of reordering constraints that are directly based on subsentential boundaries defined by (1) shallow syntactic phrases and (2) predicate argument structures.

We experiment with English-Arabic translation, which present reordering challenges. For example, English and Arabic differ in subject-verb order which might require complex long range reorderings that are more problematic for phrase-based SMT

2 CONVENTIONAL DISTANCE-BASED REORDERING

In the Moses phrase-based SMT decoder, beam search is used to find the highest scoring translation hypothesis for a given input sentence. At each step of the search, current translation hypotheses are expanded by adding the phrasal translation of a sequence of input words. Reordering is made possible by covering input phrases out of their original order, but discouraged by incorporating a reordering cost to the translation candidate scoring.

Using notations from Koehn et al. [10], an English translation hypothesis e for a French sentence f is scored as follows:

$$p(e|f) = p_{LM}(e) \prod_{i=0}^l \phi(f_i|e_i) \alpha^{|a_i - b_{i-1} - 1|}$$

Where:

- f_i and e_i are an aligned French-English phrase pair.
- $\phi(f_i|e_i)$ is the phrase-table translation score for the given phrase pair.
- α is the reordering (or distortion) weight which is optimized automatically.
- a_i is the start position of the French phrase f_i that was translated into the i^{th} English phrase e_i .
- b_{i-1} is the end position of the French phrase f_{i-1} that was translated into the $i-1$ English phrase e_{i-1} .

This model therefore prefers local to long-distance reorderings. If the translation is monotone, no phrases are reordered and the reordering cost is zero. Note that the phrase-pair score $\phi(f_i|e_i)$ can incorporate lexicalized reordering models [22] in addition to phrase-table scores. Given a French phrase f_i and its translation e_i , the lexicalized reordering model provides a probability distribution over possible positions of the next phrase to be translated with respect to f_i . While lexicalized reordering models improve translation quality [11], reordering is still performed without explicit knowledge of meaningful sub-sentential units.

In practice, an additional reordering constraint is required when building translation hypothesis: the reordering or distortion limit DL imposes a hard limit on the maximum number of input words $a_i - b_{i-1} - 1$ that can be skipped. Limiting reordering distance to DL prunes the decoding search space, and has been found in practice to reduce decoding time while improving translation quality.

In most current SMT systems, distance-based reordering models and constraints therefore have no knowledge of the input sentence structure, and can allow many incorrect reorderings. Current distance-based reordering models might break constituents and incorrectly change the meaning of a sentence, as in the following example for English to Arabic translation: (please note that we present all the Arabic examples using Buckwalter transliteration)

In Putin is **the first Russian president** to visit Turkey since Nikolai Podgorny in 1972.
Ref bwtyn hw **Awl r}ys rwsy** yzwr trkyA mn* nykwlAy bwdgwrny EAm 1972.
Hyp **Alr}ys Alrwsy** flAdymyr bwtyn **Awl** nykwlAy podgorny yzwr trkyA mn* EAm 1972.

Conversely, long-distance reorderings that are necessary to build a correct translation might be penalized by distance-based reordering models, or even prohibited by the distortion limit constraint.

3 REORDERING CONSTRAINTS FOR SUBSENTENTIAL CONSTITUENTS

Instead of relying only on the distortion limit heuristic as reordering constraint, we propose to define linguistically motivated units that cannot be broken by reordering. In other words, when building partial translation hypotheses at decoding time, all the words within the unit boundaries have to be covered before translating out-of-boundaries phrases.

For instance, assume that the phrase “the first Russian president” is a non-breakable unit for reordering in the following example sentence. This constraint prevents the system from generating incorrect translation hypotheses as in the example from Section 2.

Let’s consider the following partial translation hypothesis which covers the English words in bold:
in Putin is the first **Russian president** to visit turkey since Nikolai Podgorny in 1972 .
partial hyp 1 Alr}ys Alrwsy

With the distance based reordering model, it is possible to extend this hypothesis by translating the input phrase “Putin”, which, despite the reordering penalty, gains high score from the language model and ultimately yields an incorrect translation:
in **Putin** is the first **Russian president** to visit turkey since Nikolai Podgorny in 1972 .
partial hyp 2 Alr}ys Alrwsy flAdymyr bwtyn

If we define reordering constraints, partial hypothesis 2 cannot be generated since it covers words outside of the unit boundaries before all the words within the unit are covered. With reordering constraints, partial hypothesis 1 can only be extended by translating phrases that contains one or more of the 2 uncovered words in the unit, which yield hypotheses with low language model scores, such as the one below:

in Putin is [**the first Russian president**] to visit turkey since Nikolai Podgorny in 1972 .
partial hyp 3 Alr}ys Alrwsy Awl

In contrast, the correct translation can be generated without breaking unit boundaries:

In Putin is [**the first Russian president**] to visit turkey since Nikolai Podgorny in 1972 .
partial hyp 4 bwtyn Awl r}ys rwsy yzwr trkyA mn* nykwlAy podgorny fy EAm 1972 .

Note that the unit boundaries for the reordering constraint are independent of the phrasal segmentation used in translation: words within a zone are not necessarily translated as a single phrase and can be reordered; input phrases that cross zone boundaries can be used in translation hypotheses without breaking the reordering constraint.

With the Moses decoder, the reordering constraints described here are easily represented using XML tags in the system input [9] which means that our approach only requires a preprocessing step to mark unit boundaries in addition to conventional Moses decoding.

4 UNIT DEFINITIONS

In this paper, we will consider two types of definitions for meaningful sentence units that are non-breakable in reordering. Shallow syntax and PropBank style shallow semantics both aim at defining sub-sentential constituents, which might be useful units for constraining reordering in another language.

We define shallow syntactic structure through the use of base phrase boundaries (BP) such as noun phrases, verb phrases, adjective phrases, etc. We do not take the phrase type into consideration; we only focus on the BP boundaries. Base phrases form small syntactic units that are coherent. They are not meaning units in the way multiword expressions are since they are not fixed, nor statistically collocational, and they are by definition typically compositional. However, their syntactic juxtaposition allows for a level of coherence defined by the dependencies among the units making up a base phrase. For example, “beautiful dress” is a noun phrase where the head noun “dress” is modified by the adjective “beautiful”.

We define shallow semantics through the use of predicate argument structure boundaries (ARG) such as ARG0, ARG1, ARGM-TMP, etc. Again, we do not take the argument class label into account in our investigation. Argument boundaries group together words that form a contiguous coherent semantic unit indicating the argument boundaries pertaining to a specific predicate in the sentence.

We explore the linguistic assumption that BP and ARG form coherent semantic constructions that should not be violated during the translation process. For instance, given a sentence: “Mary bought red ripe apples that were delicious from the farmers’ market on Monday morning.” Since “red ripe” are modifiers of “apples”, they should be associated only with “apples” in the translation process rather than being associated with “Mary” or “market”, for example. Using BP reordering constraints, “red ripe apples” is considered a unit in its entirety. In our same example, “red ripe applies that were delicious” is an ARG1 of the predicate “bought”, respecting the ARG boundaries should make sure that “that were delicious” is not associated with “farmers” or “market” in the translation process.

5 RELATED WORK

We focus on related reordering approaches in the context of flat phrase-based SMT models, and will not discuss the reordering strategies specific to structure and syntax-based approaches.

In order to compensate for weak reordering in phrase-based models, sentence restructuring strategies have been proposed, where language-specific rules are applied to the full syntactic parse of the input sentence so that its word order becomes closer to that of the output language: Collins et al. [3] and Wang et al. [23] obtain improvements in translation quality by applying a small set of manually defined rules to German and Chinese sentences, while Xia and McCord [24] and Habash [8] automatically learn restructuring rules from word aligned parallel sentences.

Other approaches attempt to better integrate reordering with decoding: Zhang et al. [25] automatically learn reordering rules based on base phrase chunks for Chinese-English SMT and use a confusion network representation to consider all possible reorderings at decoding time. Elming [6] uses lattices to represent input reorderings from learned rules and integrates the cost of the reordering rules in the scoring of translation hypotheses during decoding. This approach improves translation quality for Danish to English [6] and English to Arabic [5] translation.

In contrast with all those approaches, we do not restructure the input sentence and do not design clause reordering rules that are specific to a given language pair. We focus instead on specifying different reordering constraints within the search, and define reordering boundaries using both base phrase chunks and PropBank predicate argument structures.

To date, predicate argument structures have received little attention in the context of SMT reordering. An exception is work by Komachi et al. [13], who used a Japanese predicate-argument structure analyzer that identifies verb, adjective and noun predicates and three categories of arguments that roughly correspond to the nominative, accusative and locative cases. Hand-written rules for each chunk type were applied to restructure Japanese input sentences and improved BLEU on the small-scale IWSLT Japanese-English translation task.

Finally, the XML markup representation for reordering constraints was recently implemented in the Moses decoder and used to improve translation quality on WMT09 tasks by using punctuation to define reordering units [9]. In this paper, we repurpose this Moses functionality for linguistically-motivated reordering constraints.

6 EXPERIMENT SET-UP

We evaluate the impact of the different reordering constraints on translation quality for English to Arabic translation task, we use a training corpus of about 3.36M parallel sentences (about 106M words on each side) using the following LDC catalogues: LDC2005E46, LDC2004E72, LDC2004T17, LDC2004T18, LDC2007E06, LDC2007E46, LDC2007E87, LDC2005E83, LDC2006E92, LDC2006E85 and part of LDC2007T08 and LDC2004E13. The Arabic side was converted to Buckwalter encoding and tokenized using MADA and TOKAN [7] into the Arabic TreeBank tokenization without any diacritics. For the English side we used basic tokenization.

The development set for tuning consisted of the first 200 sentences of the multiple translation Arabic test set from the NIST MT02 evaluation. We used the first English reference translation (sysid="ahd") as our input and the Arabic source as our single reference. The system performance is measured on the two test sets NIST MT04 and MT05. Similar to the development set, we used the first reference translation as our input to the system and the source as our single reference.

A. Translation system

We use the Moses phrase-based statistical machine translation system [12] and follow standard training, tuning and decoding strategies.

The translation model consists of a standard Moses phrase-table with lexicalized reordering. Bidirectional word alignments obtained with GIZA++ are intersected using the grow-diag-final-and heuristic. Translations of phrases of up to 7 words long are collected and scored with translation probabilities and lexical weighting. In addition to the standard distance-based reordering model, we trained and used a lexicalized reordering model.

The language model is a 3-gram model built with the SRI language modeling toolkit [19] using the Chen and Goodman’s modified Kneser-Ney smoothing.

The log-linear model feature weights were learned using minimum error rate training (MERT) [14] with BLEU score [15] as the objective function. Note that the weights are learned for a standard Moses system with a distortion limit of 6, and those settings are used for all the experimental conditions.

TABLE I
TEST SET STATISTICS

Test Set	Size		BP		ARG	
	sent.	Words	nb.	avg. length	nb.	avg. length
MT04	1353	46613	18233	2.9	6898	9.0
MT05	1056	35536	13860	2.9	5615	9.0

B. Linguistic preprocessing

TreeTagger [17] is used to perform POS tagging and Base Phrase chunking with standard English parameters. Constrained reordering zones are defined for all chunks of more than one word. We do not exploit the chunk labels in our experiments; we only use the chunk boundaries. The average length of the resulting zones is about two words as can be seen in Table I.

Argument boundaries are obtained by running SwiRL [20], which performs semantic role labeling on top of full syntactic analyses provided by the Charniak parser. Syntactic constituents are mapped to semantic arguments, and simple heuristics are

used to identify semantic arguments that span more than one syntactic constituent. This approach yielded competitive results on the full SRL task (argument boundary detection and classification) at CoNLL-2005 [21]. We define a constrained reordering zone for every argument. Note that arguments are tagged for every predicate identified in a given sentence, which can yield nested argument structures and therefore nested reordering zones. We only use the argument boundaries without exploiting the argument labels in this investigation. The average length of the resulting zone is about 9 words (see Table I).

7 RESULTS AND DISCUSSION

We consider four decoding conditions: unconstrained reordering (NONE), conventional default distortion limit of 6 (DIST), base-phrase based reordering (BP) and argument based reordering (ARG). Tables II and III report the impact on translation quality on the four most commonly used evaluation metrics: NIST [4], BLEU [15], METEOR [1] using exact matching, and the TER [18]. Note that all these scores are computed using only a single reference translation for all tasks, which makes the evaluation particularly harsh; however we focus here on relative scores.

TABLE III
ENGLISH TO ARABIC TRANSLATION QUALITY EVALUATION

Test set	constraint	NIST	BLEU	METEOR	TER
MT04	NONE	7.09	26.16	47.24	57.15
	DIST	7.13	26.54	47.67	55.47
	BP	7.13	26.59	47.49	56.27
	ARG	7.11	26.37	47.38	56.58
MT05	NONE	7.64	30.68	51.33	51.31
	DIST	7.69	31.18	51.62	50.21
	BP	7.69	31.22	51.56	50.58
	ARG	7.68	31.07	51.50	50.74

According to all four metrics, all three reordering constraints improve over the translation quality of the unconstrained system. Comparing BP and ARG constraints to DIST reveals that BP constraints typically yield scores that are very close and sometimes better than the DIST constraint, while ARG constraints yield slightly lower scores.

A. Sentence level results

While the differences in overall scores are small, a finer-grained sentence level analysis shows that the different reordering constraints impact translation quality. We rely on METEOR scores since they exhibit higher level of correlation with human judgments on short segments than the three other metrics considered [16].

We find that BP and ARG respectively improve on the DIST constraint for 20% and 18% of the test sentences, respectively. Interestingly, BP and ARG both improve on only 11%, which means that the BP constraints help where ARG doesn't for 9% of test sentences, while the ARG constraints help where BP doesn't for 7% of the test sentences. This suggests that the BP and ARG constraints capture different phenomena and might be complementary.

However, it should be noted that the automatic evaluation metrics used here are not very good at capturing differences in word order. Sentences with vastly different word orders can have the same BLEU score [2]. In order to get a better understanding of the impact of each of the reordering constraints, it is therefore necessary to conduct a manual error analysis to better understand the impact of the different constraints.

B. Manual analysis

We compared the impact of the three reordering constraints on a random sample of 60 sentences where at least one of the constraints yields a translation that differs from the baseline unconstrained system. This analysis shows that despite the fact that the DIST, ARG and BP constraints yield automatic scores that are very close, the ARG and BP constraints each capture different patterns that improve on DIST.

The BP constraints often help by defining shorter reordering units than the 6 word DIST constraint and the longer argument boundaries. In particular, the BP constraint was found to improve translations by not breaking nouns and their modifiers within noun phrases. In the example below, the BP constraints help translate the units "412 moroccan prisoners" and "polisario front prisons" correctly, while all other constraint types didn't:

In in its statement , the organization called for the release of **412 moroccan prisoners** of war held in **polisario front prisons** in western sahara , the former spanish colony that demands independence , which was annexed to morocco in 1975 .

Ref w dEt AlmnZmp fy byAn hA Aly ATIAq srAH **412 Asyr Hrb mgrby** mEtqlyn fy **sjwn jbhp AlbwlysAryw** fy AlSHrA' Algrbyp w Alty tTAlb b AstqlAl h*h AlmstEmrp AlAsbAnyp AlsAbqp Alty Dm hA Almgrb EAm 1975 .

NONE w fy byAn h , w dEt AlmnZmp Aly ATIAq srAH **Asry AlHrb AlmgArbp jbhp AlbwlysAryw 412** AlmEqwdp fy **Alsawn** fy AlSHrA' Algrbyp , AlmstEmrp AlAsbAnyp sAbqA An tTAlb b AlAstqlAl , Al*y Arfq b h Aly Almgrb fy 1975 .

DIST fy byAn hA **412** Aly ATIAq srAH **Asry AlHrb AlmgArbp jbhp AlbwlysAryw fy Alsawn** fy AlSHrA' Algrbyp , AlmstEmrp AlAsbAnyp sAbqA An tTAlb b AlAstqlAl , Al*y Arfq b h Aly Almgrb fy 1975 .

BP-in in [its statement] , [the organization] called for [the release] of [**412 moroccan prisoners**] of war held in [**polisario front prisons**] in [western sahara] , [the former spanish colony] that demands independence , which [was annexed] to morocco in 1975 .

BP w fy byAn h , w dEt AlmnZmp Aly AlAfrAj En **412 Asry AlHrb AlmgArbp fy sjwn jbhp AlbwlysAryw** fy AlSHrA' Algrbyp , AlmstEmrp AlAsbAnyp sAbqA An tTAlb b AlAstqlAl , Al*y Arfq b h Aly Almgrb fy 1975 .

ARG-in [in its statement] , [the organization] called [for the release of 412 moroccan prisoners of war held in polisario front prisons in western sahara , the former spanish colony that demands independence , which was annexed to morocco in 1975] .

ARG w fy byAn h , w dEt AlmnZmp Aly ATIAq srAH **Asry AlHrb AlmgArbp jbhp AlbwlysAryw 412 fy Alsawn** fy AlSHrA' Algrbyp , AlmstEmrp AlAsbAnyp sAbqA An tTAlb b AlAstqlAl , Al*y Arfq b h Aly Almgrb fy 1975 .

However, the BP constraints hurt translation quality when long nested noun phrases occur. The current BP constraints only consider the boundaries of the shorter noun phrases and reorder them incorrectly. Note that these long nested phrases tend to be incorrectly translated with the DIST constraint too.

ARG constraints help when the two other constraints don't by preventing incorrect reordering that fall within the 6 word distortion limit or that cannot be captured within a single base phrase. Also ARG captures long distance relations better. In the following example, ARG produces a more fluent and accurate translation as noted by the shared sequence of words between ARG out and ref. In the BP/DIST cases, the words are not relatable to each other since the words are not in the correct syntactic order interpreting "technology" as the head noun hence yielding the translation equivalent of word salad, namely, "the-technology the-military the-cooperation between the-two-sides". ARG captures the internal word dependencies yielding a more correct and coherent translation.:

In military technology cooperation between the two sides is being continuously increased .

Ref w ytm twsyE w dFE AltEAwn Altknwlwjy AlEskry byn AljAnbyn b Sfp mstmpr .

NONE AltknwlwjyA AlEskryp w yjry bAstmrAr AltEAwn byn AljAnbyn .

DIST AltknwlwjyA AlEskryp AltEAwn byn AljAnbyn yjry tzAyd mstmpr .

BP-in [military technology cooperation] between [the two sides] [is being continuously increased] .

BP AltknwlwjyA AlEskryp AltEAwn byn AljAnbyn yjry tzAyd mstmpr .

ARG-in [military technology cooperation between the two sides] is being continuously increased .

ARG w yjry bAstmrAr AltEAwn Altknwlwjy AlEskry byn AljAnbyn .

While the large majority of differences in SMT output with the different constraints are word order differences, we also observed few instances where the ARG constraints indirectly improved phrasal lexical choice.

8 CONCLUSION

We have compared two linguistically motivated reordering constraints for phrase-based SMT with the conventional distance-based reordering constraint. On the one hand, automatic evaluation metrics show that there is little difference in scores between the distance-based reordering constraints and the linguistically motivated base-phrase and argument boundaries constraints. On the other hand, manual analysis indicates that constraints that are tighter than the typical 6-word distortion limit are useful, as showed by the examples of improvements with base-phrase chunks constraints, while argument boundaries help when base-phrase chunk constraints fail to capture long nested noun phrases.

REFERENCES

- [1] Satanjeev Banerjee and Alon Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgement," in *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan, June 2005.
- [2] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. "Re-evaluating the role of BLEU in machine translation research." In *Proceedings of 11th meeting of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 249–256, April 2006.
- [3] Michael Collins, Philipp Koehn, and Ivona Kucerova. "Clause restructuring for statistical machine translation." In *Proceedings of ACL 2005 (Meeting of the Association for Computational Linguistics)*, pages 531–540, 2005.
- [4] George Doddington. "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics." In *Proceedings of the Human Language Technology conference (HLT-2002)*, San Diego, CA, 2002.
- [5] Jakob Elming and Nizar Habash. "Syntactic reordering for English-Arabic phrase-based machine translation." In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69–77, Athens, Greece, March 2009.
- [6] Jakob Elming. "Syntactic reordering integrated with phrase-based SMT." In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 46–54, Columbus, Ohio, 2008.

- [7] Nizar Habash, Owen Rambow, and Ryan Roth. "MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization." In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, Egypt, 2009.
- [8] Nizar Habash. "Syntactic preprocessing for statistical machine translation." In *Proceedings of the Machine Translation Summit (MT-Summit)*, Copenhagen, 2007.
- [9] Philipp Koehn and Barry Haddow. "Edinburgh's submission to all tracks of the WMT 2009 shared task with reordering and speed improvements to Moses." In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 160–164, Athens, Greece, March 2009.
- [10] Philipp Koehn, Franz Och, and Daniel Marcu. "Statistical phrase-based translation." In *Proceedings of HLT/NAACL-2003*, Edmonton, Canada, May 2003.
- [11] Philipp Koehn, Amitai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. "Edinburgh system description for the 2005 IWSLT speech translation evaluation." In *Proceedings of IWSLT-2005*, Pittsburgh, PA, 2005.
- [12] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. "Moses: Open source toolkit for statistical machine translation." In *Annual Meeting of the Association for Computational Linguistics (ACL)*, demonstration session, Prague, Czech Republic, June 2007.
- [13] Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. "Phrase reordering for statistical machine translation based on predicate-argument structure." In *Proceedings of the International Workshop on Spoken Language Translation*, pages 77–82, Kyoto, Japan, November 2006.
- [14] Franz Josef Och. "Minimum error rate training in statistical machine translation." In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, 2003.
- [15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.
- [16] Mark Przybocki, Kay Peterson, and Sébastien Brossard. "Official results of the NIST 2008 metrics for machine translation challenge (metricsmatr08)." Technical report, NIST, 2008.
- [17] Helmut Schmid. "Probabilistic part-of-speech tagging using decision trees." In *Proceedings of the Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.
- [18] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. "A study of translation edit rate with targeted human annotation." In *Proceedings of AMTA*, pages 223–231, Boston, MA, 2006. Association for Machine Translation in the Americas.
- [19] Andreas Stolcke. "SRILM: an extensible language modeling toolkit." In *International Conference on Spoken Language Processing*, Denver, Colorado, September 2002.
- [20] Mihai Surdeanu (2007), SwiRL, available from: <http://www.surdeanu.name/mihai/swirl/> (accessed 1 Sep 2011)
- [21] Mihai Surdeanu and Jordi Turmo. "Semantic role labeling using complete syntactic analysis." In *Proceedings of the 9th International Conference on Computational Natural Language Learning (CoNLL)*, Ann Arbor, MI, 2005.
- [22] Christoph Tillmann. "A unigram orientation model for statistical machine translation." In *HLT-NAACL 2004*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.
- [23] Chao Wang, Michael Collins, and Philipp Koehn. "Chinese syntactic reordering for statistical machine translation." In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 737–745, 2007.
- [24] Fei Xia and Michael McCord. "Improving a statistical MT system with automatically learned rewrite patterns." In *Proceedings of COLING 2004*, pages 508–514, Geneva, Switzerland, August 2004.
- [25] Yuqi Zhang, Richard Zens, and Hermann Ney. "Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation." In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting*, Rochester, NY, April 2007.

An Enhanced Method for Ranking Arabic Web Pages Using Morphological Analysis

Esraa A. Hamed ^{*1}, Nagwa L. Badr ^{**2}, Mohamed F. Tolba ^{*3}

**Scientific Department, Faculty of Computers and Information Sciences*

Ain Shams University, Abassia, Cairo, Egypt

¹esraa_raoof@cis.asu.edu.eg

³fahmytolba@gmail.com

*** Information System Department, Faculty of Computers and Information Sciences*

Ain Shams University, Abassia, Cairo, Egypt

²nagwa_badr@hotmail.com

Abstract— Recently, Search engines have become one of the most important tools, due to the web browsing most useful techniques. As an essential tool for fulfilling the user query, page ranking developed for the large continuous dynamically growing number of web pages that search engine databases contain. However, the search engines ranking technique faces many challenges; one of them is ranking Arabic web pages. Information is retrieved through today's existing search engines based on the exact match with an Arabic query regardless of the morphological variations of Arabic words. Nevertheless, two words in Arabic language could have the same letters with different meanings. The existing search engine algorithms usage will end up retrieving irrelevant information for users. In this paper, we propose a new ranking technique based on morphological meanings of Arabic words combined with the web link structure of which its ranking structure is based on counting the words that are related to the query in relevant documents and its outgoing links. Moreover, we enhance the Arabic ranking module by parallelizing its components to be more scalable at certain cases. Finally, we evaluate the accuracy of our ranking technique by performing experiments using real-world data, further more we evaluate the efficiency of its parallelization and it proved success.

1 INTRODUCTION

At present there are roughly around **56** million Arab internet users in the Arab world, representing only **17%** of the **337** million populations. The number of Arabic internet users in the Middle East and North Africa is expected to grow by nearly 50% over the next three years, rising to 82 million users by 2013 [1].

There are many challenges in building good search engines. One of these challenges is the continuous and rapid growth of the web. The growth rate of the web is even more dramatic. According to latest statistics [2, 3], the size of the web has doubled in less than two years.

Search engines have main five components [4], a crawling module for downloading web pages [13], an indexing module for generating a lookup table for the downloaded pages, a page repository for containing a local copy of the downloaded pages, a query engine for fulfilling the user queries, and a page ranking module for sorting the search results.

The page ranking module is responsible for sorting the results such that results near the top are the most likely ones to be what the user is looking or on hyper-link. We focused on implementing an enhanced ranking algorithm by combining both the page content and the Hyper-Link. Moreover we focused on Arabic search engines using the morphological meaning of the Arabic word database having the morphological meanings of the most Arabic words.

The proposed method rank is more efficient than ranking using other engine not considered the morphological meaning aspect. The enhancement of the Arabic ranking module by parallelizing its components is essential because the size of the web grows at a remarkable speed and centralized page ranking is not scalable. The achievement of the best speeding up needs to determine two issues: the first issue is to achieve high scalability, and the second one is how many processors required for achieving such kind of parallelization.

This paper is organized as follows. In section 2, presents background about the research point. Section 3 highlights the proposed architecture. Section4 discusses the implemented prototype. Section 5 evaluates that technique. Section 6 concludes the paper with some predictions.

2 PAGE RANKING TECHNIQUES

Many of Search engines use a traditional text process to retrieve pages related to a user's query. Traditional text processing is trying to find all documents using the query terms, or related to the query terms by semantic meaning. With the massive size of the web, this step can result in thousands of retrieved pages related to the query.

The main function of the ranking module is to sort the search results by relevance or importance using information retrieval (IR) algorithms. On the other hand, The Web is much less coherent, changes more rapidly, and is spread over geographically distributed computers. So Traditional text processing can't filter sufficient numbers of irrelevant pages out of search results thus, link analysis has become the means to ranking. Each page/document on the Web is represented as a node in a very large graph. The directed arcs connecting these nodes represent the hyperlinks between the documents. This hyperlink structure is exploited by three of the most frequently cited Web IR methods: HITS (Hypertext Induced Topic Search) [5], PageRank [6] and SALSA (Stochastic Approach for Link Structure Analysis) [7].

Recent studies showed that non-English queries and unclassifiable queries have nearly tripled since 1997. As they do not take full account of significant features of languages which are absent or unimportant in English. Such features include using of capitals in individual languages [10].

A morphology system is the backbone of a natural language processing system. No application in this field can work without support of a good morphology system. The Arabic language has its own features that is why a lot of research effort in this area [8]. Search quality is measured by two factors; Recall and Precision .Without using the morphological analyzer there will be poor "recall" and high" precision". In English language you won't find similar poor results because prepositions are separate words. So to increase the "Recall" while searching in the Arabic full-text, you have to use the morphological analyzer [9].

The main conclusion from literature is that searching using non-English and non-Latin script queries results in lower success and requires additional user effort to achieve acceptable recall and precision. Further international search engines are relatively inaccurate with monolingual non-English queries [10].

As the size of the web grows, it becomes more difficult, or impossible, to rank the entire web by a single process. Distributed page ranking are needed because the size of the web grows at a remarkable speed and centralized page ranking is not scalable.

This research addressed the problem of enhancing the performance of language specific page rankings. As parallel rankers are challenging to operate, however they have important advantages, compared to single-process rankers like; scalability, the speedup of the program that tells you how much performance gain is achieved by running your program in parallel on multiple processors, and efficiency that defined as the ratio of speedup with p processors to p [12].

3 THE PROPOSED ARCHITECTURE OF THE ARABIC WEB RANKING TECHNIQUE

This research addresses a new method for ranking the Arabic web sites which are based on morphological meanings of Arabic words. The Work proposes an enhanced ranking algorithm by combining both the count of words related to query in the page and outbound pages of that page. Furthermore by using external database that having the morphological meanings of the most Arabic words. Then it sorts the pages according to its rank.

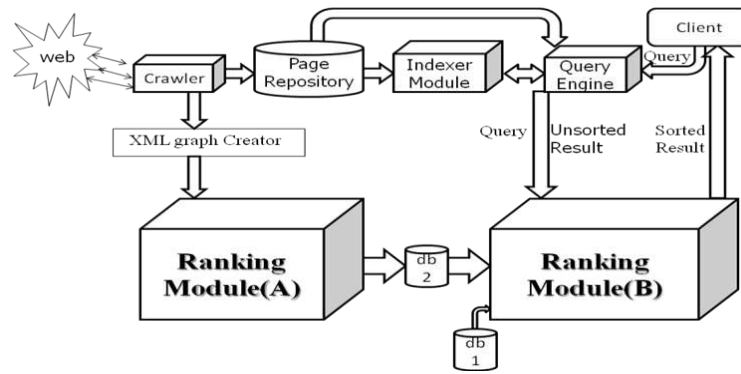


Figure 1: The proposed Architecture of the Arabic Web Ranking Technique.

In the proposed Architecture (see figure 1) for the Arabic Ranking Module, the Crawler is downloading the web pages. While crawler was crawling the web, the XML graph creator created xml file that have each downloaded page.

The Indexer module reads the repository and generates a lookup table with all the URLs that point to pages that contain a given word (the text index). When a user enters a query into a search engine (typically by using keywords), the Query engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text.

The ranking module sorts the results such that results near the top are the most likely ones to be what the user is looking for. There were two kinds of methods in information retrieval, based on both content and based on hyper-link. Because that the quantity of computation in systems based on content was very large and the precision in systems based on hyper-link only was not ideal.

In this research, we focus on implementing an enhanced ranking algorithm by combining both the Content Information and the Hyper-Link with the focus on Arabic search engines by taking into account the stem and the context of the Arabic word. This combination of Link and Content Information proofed it successful [15] and achieving high quality of the best search but on the English –based search only, however the Arabic search still not covered.

In our proposed ranking technique, we have two modules. The Ranking Module (A) works offline, as it receives the XML file graph that has all downloaded pages from the crawler. Then it creates database (db2) that has each downloaded document with the list of all Arabic words and its frequency that exist in this document and in its outbound pages. The detailed processing steps of Ranking Module (A) are illustrated as follows:

- Module (A) receives the XML file graph from the crawler after XML graph creator creates it.
- Read documents and its outbound documents and divide for each into words and storing all words in the string array with eliminating the stop words, and list of non Arabic litters.
- Create a database that has each downloaded document with all Arabic words and its frequency in this document and in its outbound pages.

Ranking Module (B) works online as it gets the query and unsorted results from the query engine to rank them using the proposed ranking technique. By accessing db1 (i.e. db1 has the morphological meanings of the most Arabic word) and db2 (i.e. db2 is created by the Ranking module (A)) then it sorts the resulting pages according to its rank and displays them to the user. The Processing steps of Ranking Module (B) are illustrated as follows:

- Receive the query and unsorted result from the query engine to rank it using our proposed ranking technique.
- Read the morphological meaning of query words from db1 which has the morphological meanings of the most Arabic words.
- If there is more than one different meaning to an input query word, the user may choose the meaning he/she wishes to search for. The search results will largely contain the inflected forms of the word that belong to that meaning. This helps reduce the redundancy that results from morphological search only.
- Save the query meanings that user chose.
- Access the db2 which created from Ranking Module (A) and has all web pages downloaded from crawler with all Arabic words with its frequency that exist in it and in its outbound pages.
- Save the words of the resulting page and compare the meaning of each word with the list of query meanings.

- Follow the following steps For each saved word :
 - Step1: Get the morphological meanings of the first word and save it in the list of meanings.
 - Step2: Compare this list of meanings of word with the list of query meanings.
 - Step3: If the word has similar meaning to one of the query meanings, weight the word according to its meanings number from 0-1 Then add to page rank the word frequency multiplied by its weight. Therefore the total page rank for each resulting page equal the weight of frequency for each word related to query which exist in the page and in its outbound pages.
- Sort the resulting pages according to its rank and show them to the user.

4 THE PARALLELIZATION FOR THE PROPOSED RANKING TECHNIQUE

This section addresses the problem of enhancing the performance of the proposed Arabic ranking technique. It is efficient ranking technique for Arabic search results, but it is relatively slow. Parallel ranking in this case for ranking module (A) has a great advantage to solve the problem of the size of the web documents.

The parallelization algorithm is illustrated as follows:

- The XML graph division receives the XML file graph from the crawler after XML graph creator creates it.
- Divides the XML file into number of XML files that equals number of processors in the ranking module (A).
- Each node in Ranking module (A) works in parallel with its own XML file to reduce the time taken as explained in the following steps:
 - Step1: Read the XML file and save parent documents in hash table as an object makes its ID number as a key.
 - Step2: Divide the document into words and storing all words in the string array.
 - Step3: Eliminate the stop words, English letters, numbers, and all non Arabic letters.
 - Step4: Create list of words and save objects that have each word with its frequency.
 - Step5: Read outbound pages, child pages of each document and check if not exist in the list of outbound pages of the parent page save it.
 - Step6: Divide each into words and eliminating also all stop words and all non-Arabic letters.
 - Step7: Save the outbound page in hash table and creates for it list of its Arabic words with the frequency of each word.
- Add the result to (db2) that collects in the master node which accessed by ranking module (B).

5 THE IMPLEMENTATION OF THE PARALLEL ARABIC RANKING TECHNIQUE

This section describes in details the experiment of the ranking technique and its parallelization. We retrieve the relevant documents for the suggested query and choose others search engines to find the relevant documents for the suggested query word. The retrieved documents for each search engine are ranked by using the proposed method which is based on the combination between Page content and link taking into account the stem and the context of the Arabic word.

The performance plus the rank results of each search engines were compared to the number of users ranking which aims to determine the best one. The two selected search engines are Google search engine, and Yahoo search engine. The pre-processing steps in order to retrieve data that are summarized as follows:

- The database collects in the master node which accessed by ranking module (B).
- Use the Arabic dictionary which is a comprehensive dictionary of contemporary Arabic (Modern Standard Arabic). It includes up-to-date words used in the various media [11].
- Write any query word in Arabic in order to retrieve the relevant Arabic documents of each query using Yahoo, and Google.
- Pick up the first thirty documents, which are retrieved by each search engine, and then the retrieved documents are saved as text documents.
- Using the existing distributed crawlers [13] retrieve for each set of documents its outbound links and save them in xml files.

Then apply the steps that mentioned before of the proposed method which is based on combination between the morphological analysis and the hyperlink structure for ranking Arabic documents for some Arabic domains.

Although, Ranking module (B) take less than one second to rank results and sort it for user, but ranking module (A) that indexes pages in the database with its Arabic words makes the algorithm relatively slow. To get an idea of how much this delay is, we applied the Ranking module (A) on a dataset of 10000 Arabic web pages [12]. To apply the parallel algorithm shown in this paper, we need a number of processors (N) working together. The algorithm for each processor is as follows:

- When the crawler download pages from the web, the XML file creator will create the XML file graph for all downloaded pages.
- The XML division divides the XML file to n XML files with the same size (ex: n=10, then each XML file has 1000 page, with size approximately 90 KB).
- The master node of module (A) distributes the XML files among the processors.
- Each processor applies the algorithm of module (A) on its own XML file.
- Each processor sends its output database to the master node.

The master node then merges all the outputs database received from the other processors with its own database and delete the redundant pages if exist.

6 EVALUATION

The performance measurements of experiment that described in the previous sections are applied on the real data and gave efficient results. These results are compared with two different search engines to highlight its effectiveness. As the speed consumption is an important issue in Ranking the Web therefore evaluating the parallelization and the speed up of the ranking process is an effective stage.

Section 6.1 show the performance results for the implementation of the proposed Arabic ranking technique, section 6.2 shows the performance results for the implementation of its parallelization.

A. Performance Results for the Proposed Arabic Ranking Technique

This approach is using Visual studio .Net 2005 software. The proposed method is based on the combination between the morphological meanings of Arabic words in the page, and outbound pages. In this section we apply it on real data to see its performance when it is compared with others. Consider two different search engines to compare between their results with the ranking method.

Measurement the effectiveness of the proposed method to other ranking methods is essential aspect. So, we chose twenty documents from each search engines, because the number of considered documents do not affect on the algorithm performance, and do not effect at the algorithm results.

- 1) *Google search engine results:* We show in table I the performance results for Google search engine. It shows comparison between the proposed method, and some interested users in the Arabic query word. It also shows comparison between Google search engine and ranking of the five interested users.

TABLE I
The proposed method ranking results against Google ranking results according to five interested users

P O S I T I O N	User Ranking					Ranking according to Google search engine	Ranking according to our proposed method	Average Error for Google search engine according to interest users					Average Error the Proposed ranking according to interest users				
	1	2	3	4	5			1	2	3	4	5	1	2	3	4	5
1	2	12	12	12	12	1	2	6	9	7	9	10	4	1	3	1	0
2	12	18	18	7	7	2	12	1	6	1	1	2	0	7	2	2	3
3	18	3	2	2	6	3	18	1	0	3	3	2	3	4	1	1	2
4	3	1	6	6	2	4	6	4	1	3	1	4	0	3	1	3	0
5	6	4	7	4	3	5	7	5	2	6	4	5	0	3	1	1	0
6	7	6	3	3	15	6	15	1	0	2	2	3	1	2	0	0	1
7	1	5	4	18	18	7	3	1	3	2	5	5	1	1	0	3	3
8	4	2	1	15	4	8	4	3	4	5	4	4	2	1	0	1	1
9	15	8	17	5	17	9	17	11	11	11	11	11	0	0	0	0	0
10	5	7	15	1	5	10	5	4	4	5	5	4	1	1	0	0	1
11	8	13	5	17	1	11	1	1	8	3	5	6	5	2	3	1	0
12	11	16	13	8	8	12	19	10	11	11	11	11	0	1	1	1	1
13	13	17	8	19	13	13	8	0	2	1	1	0	1	3	2	0	1
14	10	10	11	13	10	14	13	3	3	2	3	2	1	1	0	1	0
15	17	20	10	10	19	15	10	6	6	5	7	9	3	3	4	2	0
16	19	19	14	11	14	16	14	3	2	2	2	3	0	1	1	1	0
17	14	15	19	14	11	17	11	2	4	8	6	8	6	4	0	2	0
18	20	14	16	16	20	18	20	15	16	16	11	11	0	1	1	4	4
19	16	11	20	20	16	19	16	3	3	2	6	4	4	4	5	1	3
20	9	9	9	9	9	20	9	2	5	1	1	2	0	3	1	1	0
The Average Error								4.1	5	4.8	4.9	5.3	1.6	2.3	1.3	1.3	1

Table I was divided into six columns, the first column named "position" addresses the position of first twenty documents that were retrieved by Google as relevant documents for the suggested Arabic query word.

The second column named "User Ranking" represents the ranking of the five interested users for these twenty documents. The third column called "Ranking according to Google search engine" represents the Google search engine ranking for the same documents.

The fourth column of the table that called "Ranking according to our proposed method" represents our proposed ranking for the same documents. The fifth column shows the distance between each document position and its correct position in user 1, user 2....user 5. The ranking in the fifth column of table that called "Average Error for Google search engine according to interest users". And the last column that called "Average Error for the proposed ranking according to interest users" represents the distance between each document position in our method ranking.

Calculating the average for the values in column 1 of the column five is to compute the average ranking error for Google search engine according to user 1. In Google ranking for every document is away from its correct position in user 1 ranking 4.1 positions on average. The same calculations for user1 are repeated for other five users, therefore, according to user 2, user 3, user4, and user5 the average ranking errors for Google search engine are 5, 4.8, 4.9, and 5.3 respectively. Also, the average ranking errors for the proposed method where is compared with Google are 1.6, 2.3, 1.3, 1.3, and 1 respectively. The average error for Google search engine ranking compared with user 1 ranking, user 2 ranking, ...and user 5 ranking is equal to $(4.1+5+4.8+4.9+5.3) / 5 = 4.82$. The average error for the proposed method is equal $(1.6+2.3+1.3+1.3+1) / 5 = 1.5$. The results show that the proposed method is better than Google search engine ranking 3.2 times.

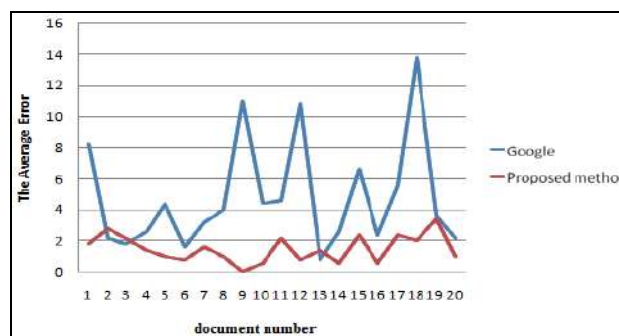


Figure 2: Google Ranking and the Proposed Method Ranking Compared with the Five Interested Users.

As shown in figure 2 the proposed method for ranking more efficient than Google according to the ranking of the five users. For example, the difference between document 12 position in Google ranking, and its position in the five users in

average is $(10+11+11+11+11) / 5 = 10.8$. The difference between document 12 position in the proposed ranking, and its correct position in the five users ranking on average is $(0+1+1+1+1) / 5 = 0.8$.

- 2) *Yahoo search engine results:* We repeated the previous test with Yahoo search engine. We repeated the previous test with Yahoo search engine. The average error for Yahoo search engine ranking compared with user 1 ranking, user 2 ranking ...and user 5 ranking is equal to 5.4. The average error for the proposed method is equal 1.26. We can conclude that the proposed ranking is better than Yahoo search engine ranking by 4.3 times. In figure 3 there are two curves, the first curve for the average difference between each document position in Yahoo ranking, and the position of each document position in the five users ranking. The second curve shows the average difference between each document position in our proposed Arabic ranking, and the position of each document position in the five users ranking.

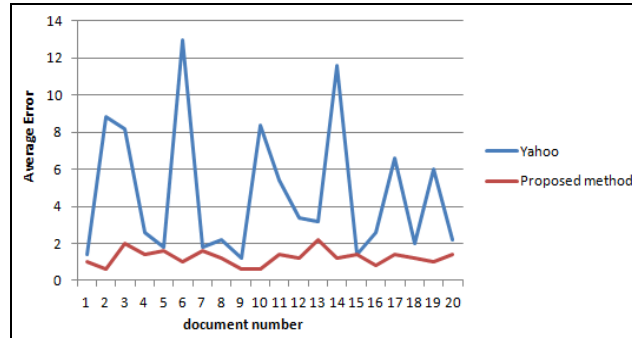


Figure 3: Yahoo Search Engine Ranking and the Proposed Method Ranking Compared with the Five Interested Users Ranking.

From the previous comparison the proposed method for ranking gives better results than Yahoo search engine according to the ranking of the five interested users. For example, the difference between document 10 position in Yahoo ranking, and its position in the five users in average is 8.4 positions. The difference between document 10 position in our proposed ranking methods and its correct position in the five users ranking on average is 0.6 positions.

B. Performance Results for a Parallelization technique

The ranking module deals with huge number of web pages, and they should maintain these pages up-to-date. Therefore, speed consumption is one important issue in Ranking the Web. In this research, parallelization technique gave efficient results for the user query as it save the Ranking time and speed up the ranking process.

To study the effect of the parallelization in this algorithm on the Ranking module speed, the proposed parallel algorithm applied on a set of about 10000 Arabic web pages. The average elapsed time (in seconds) described in the three following main stage, the stage of reading the XML file and dividing it into equal or semi-equal pieces, the ranking applied by each processor, and the merge stage of database applied by the master node.

The time spent by each processor in the Ranking module (A) stage is different from the times spent by the others. The master node should wait until all processors finish their work and send their results to this node in order to begin the merge stage. Also, we neglected both the distributing time over the processors and the time of gathering results from these processors into the master node, because they are relatively small when compared with the time of any stage of these three stages.

When the load is distributed over large number of processors, the speed of ranking increased. Also in this case, each processor takes small number of pages so, it's expected that if the number of processors N increases, the time of the Ranking module (A) stage decreases.

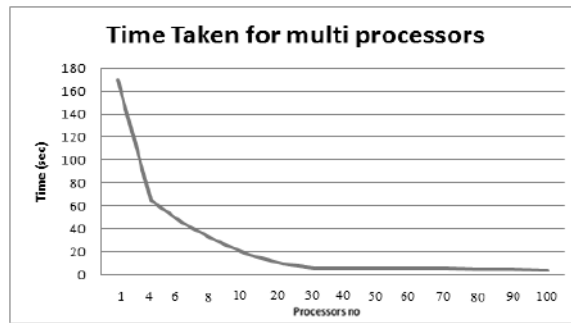


Figure 4. The Elapsed Time against the Number of Processors.

Figure 4 shows the time elapsed in ranking with the number of processors. It's shown that the time of the Ranking decreases when N increases, until N reaches ten the time start to be almost constant. This means that the optimal N for this stage is ten, because there is no significant gain in the speed can be obtained by increasing the number of processors.

We compute the speedup of the Ranking until ten processors, where *speedup* of the program shows how the performance is increased by running the program in parallel on multiple processors. A *speedup* is defined as the length of time it takes to run on a single processor, divided by the time it takes to run on a multiple processors.

$$S_n = T_{p=1} / T_{p=n} \quad [12]$$

Speedup generally ranges between 0 and p, where p is the number of processors. Computing speedup is a good way to measure how a program scales as more processors are used. [12].

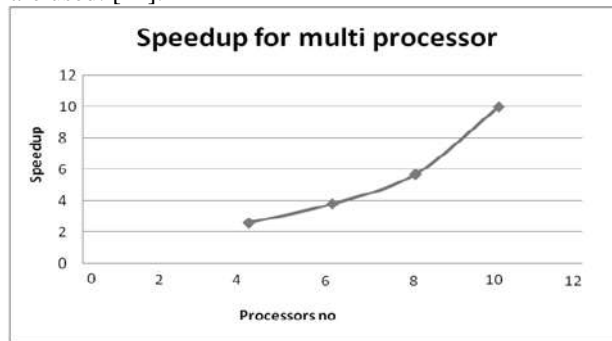


Figure 5: Speedup versus the Number of Processors.

Figure 5 shows the speedup in the Ranking module (A) with the number of processors. Efficiency is a measure of parallel performance that is closely related to speedup and is often also presented in a description of the performance of a parallel program. Efficiency with p processors is defined as the ratio of speedup with p processors to p.

$$E_p = S_p / P \quad [12]$$

Efficiency is a fraction that usually ranges between 0 and 1. $E_p=1$ [12] corresponds to perfect speedup of $S_p=p$. You can think of efficiency as describing the average speedup per processor [12]. Figure 10 shows the Efficiency of Ranking with N number of processors.

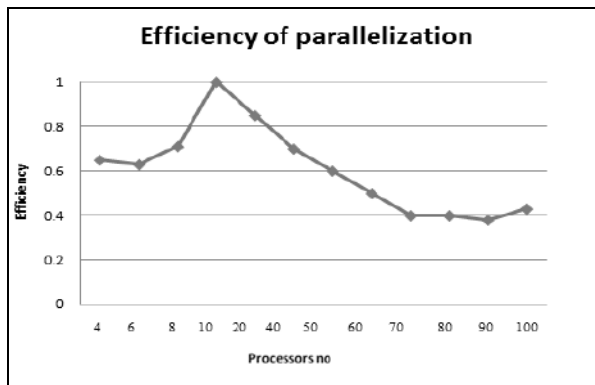


Figure 6: The Efficiency against number of processors.

From figures 5 and 6 we concluded that we got perfect speedup at number of processors $N=10$ because the efficiency of the algorithm when using ten processors equals one.

7 CONCLUSION

Through this work, we display an enhanced ranking algorithm for Arabic web pages, considering a parallelization technique. The proposed algorithm combines both count of words related to query which exist in the page and outbound pages. The algorithm uses an external database that contains the morphological meanings of the most Arabic words, and then sorts pages according to its rank. The proposed algorithm for ranking is more efficient than engines not utilizing morphological meaning as shown in the result.

In addition, a parallel technique is proposed to enhance the performance of the modified algorithm. It's obvious from the results that the Ranking time decreases when the number of processors is decreased until reaching ten, the moment when time starts to be almost constant. This algorithm performance is proved to be more efficient compared to other search engines not considering the morphological meaning.

REFERENCES

- [1] V. Cerf, Chief Internet Evangelist, Google Web Site: <http://www.startuparabia.com/page/10/>, (accessed Feb 17, 2010).
- [2] S. Lawrence, C. L. Giles, "Searching the World Wide Web," *Journal of the Science*, no.280, pp. 98-100, 1998.
- [3] S. Lawrence, C. L. Giles, "Accessibility of information on the web," *Journal of the Nature*,no. 400, pp. 107-109, 1999.
- [4] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, S. Raghavan, "Searching the Web," *ACM Transactions on Internet Technology*, no.1,pp. 2-43, 2001.
- [5] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *ACM*, no.46, pp.0004-5411, New York, NY, USA, 1999.
- [6] S. Brin, L. Page, R. Motwami, T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Technical report, Computer Science Department, Stanford University, 1999.
- [7] R. Lempel, S. Moran, "The stochastic approach for link-structure analysis (SALSA) and the TKC effect," In *The Ninth International WWW Conference*, 2000.
- [8] S. Abuleil ,K. Alsamra, "New Technique to Support Arabic Noun Morphology: Arabic Noun Classifier System (ANCS)," *International Journal of Computer Processing of Oriental Languages*, no.17, pp. 97-120, 2004.
A. Hammad, Arabic Morphological Analysis, Web Page: <http://sites.google.com/site/ahammad/arabic-morphological-analysis>. (accessed 3 March 2010).
- [9] F. Lazarinis, J. V. Ferro, J. Tait, "Improving Non-English Web Searching," *ACM SIGIR*, no.41, 2007.
- [10] H. Mahgoub(2006), Available from: <http://www.alkhwarizmy.com/En/ksearch.html>, (accessed 5 March 2010).
- [11] S. Shi, J. Yu, G. W. Yang, D. X. Wang, "Distributed Page Ranking in Structured P2P Network," *International Conference on Parallel Processing*, pp.179, 2003.
- [12] S. M. Sadjadi. "Parallel Computing Explained Parallel Performance Analysis", Web Site: <http://ci-tutor.ncsa.uiuc.edu/>, (accessed 20 March 2009).
- [13] D. Ezzat, M. Abdeen, M.F. Tolba, "A memory efficient approach for crawling language specific web: The arabic web as a case study," *Proceedings International Conference on Information Management and Engineering ICIME 2009*, Kuala Lumpur.

Persian Morphology: Description and Implementation

Vahid R. Mirzaeian*¹

**ELT Department, Faculty of Engineering, Iran University of Science and Technology
Farjam Street, Narmak, Tehran, Iran*

¹mirzaeian@iust.ac.ir

Abstract— This paper is dedicated to the linguistic and computational description of Persian morphology based on CG and HPSG formalisms. First a theoretical background on CG will be established, later morphological issues pertaining to some major aspects of Persian morphology will be explored respectively.

1 INTRODUCTION

Our treatment of Persian morphology is based on CG [1]. In this formalism, our lexicon (the dictionary file to be specific) contains a set of stems and affixes, with the process of combining the two mediated by a set of morphotactic rules based on Categorical Morphology which operate in much the same way as the classical 2-level rules of Koskiennemi. [2]. All the necessary information is included in the definition of the stems and affixes. For instance, stems of verbs in Persian normally subcategorize for different kinds of affixes such as tense markers and agreement markers. These affixes have to appear in a given order. The stems have a subcategorization requirement which contains the feature structure of the affix the stems need in order to form sublexical signs of the next higher order. This sublexical sign in turn subcategorizes for an affix of a certain kind. This process continues until the sign requires no more affixes and thus is morphologically saturated.

HPSG-based systems including ours normally rely on the interaction of two distinct sets of information: firstly, a dictionary storing information about lexical signs, and secondly, a set of rules and principles governing the possible combinations of lexical signs available. It will be more efficient to store morphemes in the dictionary and extend the set of governing rules and principles in such a way that they cover not only the possible combinations of lexical signs within sentences, but also the combination of smaller linguistic units to create lexical signs. Although this approach requires an extended set of grammatical rules because morphological rules have to be added to the syntactic rules, it has advantages in at least two respects: the dictionary will be more concise and consequently more easily maintainable, and it can be shown the grammatical information lexical signs have is rule-governed to a large extent. It has to be noted that the formation of morphological structures can be described with a few categorial rules which the system uses. Some of these rules will be discussed later in this article.

2 MORPHEM INVENTORY

Morphemes, the smallest grammatical units in the language, can be defined as sublexical signs within HPSG formalism. Sublexical means that it is a sign which is smaller than a word and can be analyzed at morphological level. As a result, these sublexical signs have syntactic and semantic features and they are treated separately in the parser as suggested by Pollard and Sag [3]. Morphological structures will be described by allowing sublexical signs to subcategorize for other such items. The subcategorisation will be captured with the help of combinatory rules (based on Categorical Morphology) which will be described below.

3 CATEGORIAL RULES

We use Steedman's slash notation [4] for the description of the morphological rules governing morphological processes in our current system. For the treatment of morphology here, in addition to ordinary association, we need Steedman's forward composition rule (3). The association rules (1 and 2) mentioned below are necessary in order to conclude morphological processes, i.e. to form a fully inflected lexical sign:

$$\begin{array}{l} X / Y \quad Y \quad \rightarrow \quad X \\ X \quad \quad \quad Y \backslash X \quad \rightarrow \quad Y \end{array}$$

Using the composition rule given below, sublexical signs can be created:

X / Y Y/Z → X/Z
 Y\Z X\Y → X\Z

4 ZERO AND EMPTY MORPHS

In his morphological analysis of German, Schulze [4] makes a distinction between zero and empty affixes. He mentions that “Morphemes with a form but no evident meaning will be referred to as empty morphs.” [4]. However, he claims that “... if there is enough evidence for meaning, but one cannot identify a corresponding form” [4] that will be called a zero morph. In our analysis, we have found the zero morph concept quite useful; therefore, we provide a typical example here:

TABLE 1
 ZERO MORPH CONCEPT

Stem	Present	Past	Infinitive	Meaning
خور xvr	خور xvr∅	خورد xvrd	خوردن xvrđn	eat

As it can be seen, in Persian regular verbs, the past as well as the infinitive form of the verb have their own affixes; however, the present stem carries no such affix. In order to change the present stem into past, a past-making affix should be added. Since “No feature set can have two different values for the same attribute” [5] and this will create potential problems especially when combining with the past affix, we have decided to use the zero affix concept illustrated by the symbol ‘∅’ Therefore, in our treatment, Persian stems are neither present nor past. If they combine with the past making affix, they will be past, if not, they will combine with a zero affix and change into present.

5 A PRACTICAL EXAMPLE

In order to see how our system deals with morphemes, let us look at a typical example from Persian. Look at the table below:

TABLE 2
 A TYPICAL PERSIAN VERB

Tense Marker	Stem	Agreement Marker	Final Form	Meaning
می My	خور xvr	م m	می خورم myxvrm	I eat

The stem خور / xvr is a sublexical sign needing a lexical affix to form a sublexical sign of the next higher level of complexity. When it finds that type of affix, (the past-making affix or present zero affix, for example) it combines with it and forms the sign of the next higher level. This sign is still unsaturated and requires another affix, namely agreement marker to combine with. This agreement marker does not have any further subcategorization requirements and shows therefore that the word is now fully inflected, i.e. morphologically saturated.

Here we want to see how we can capture the above construction morphologically within our current parser. First we have added the following entry for the stem خور / xvr in the dictionary:

```
"xvr" $$ X lectype verb (d) delayed vtype(X, valency(2, [agent, ra])) :-  
    verb(X).
```

Before we move any further, it is necessary to elaborate on the entry above.
lectype verb(d)

We have classified Persian verbs according to their properties in a file called lectype.pl. For instance (d) here means there are verbs in Persian that fall into this category, namely the present stem can be changed into the past stem by the addition of the د / d morpheme¹.

```
delayed vtype(X, valency(2, [agent, ra]))
```

Since the information we give the system for each sign tends to be bigger and this may slow the system to a great extent, we force the system not to execute this part of the code until it gets to the stage where it knows more about the lexical sign involved. This piece of code will be executed later.

```
vtype
```

This term specifies what arguments we need for that particular verb, how many of them are obligatory and finally what their thematic role and syntactic type are. This is the material from which we derive subcat frames². The simplest verbs are specified by something like

```
vtype(X, valency(2, [agent, object]))
```

The list says what roles the arguments have to play. If it is just a list of atoms, we assume that they are all expected to be NPs. The rule says that the first N element of this list is obligatory; therefore if we wrote:

```
vtype(X, valency(1, [agent, object]))
```

the object would be optional. So for the verb open in English, for example, we can write:

```
vtype(X, valency(1, [object, agent, instrument])).
```

This rule will give us the following sentences:

“John opened the door with the key.”

“John opened the door.”

“The key opened the door.”

“The door opened”.

Let us return to our discussion. The information about the stem خور / xvr will be added to our database; however, before it is added, the system will check the lexical type of this entry to see what requirements the word has got. If there is no mention of such information, the item will be treated as default. The system, then, will check the default file called lexdefau.pl³ to see what default configuration should be imposed on the item in question. Below, you can see a rule in that file saying that all items by default require a first affix called *1:

```
needsFirstAffix(X) :-  
    affixes@X <-> [A1],  
    affix@A1 <-> *1,  
    [lex_type, branch, syntax, uses]@X  
    <-> [lex_type, branch, syntax, uses]@A1.
```

Now, we will parse this item to see how our system treats this stem:

```

Sign(morph(affixes(["VERB4"(D)]),
  -affix,
  history(-umlauted, branch([], E),
  verb(d)),
  syn(nonfoot(head(cat(xbar(+v, -n)),
    vform(vfeatures(finite(tensed(F), G),
      -aux,

      subcat(args([ ... ]), fixed(T)),
      foot(wh([]), topicalised([]))),
  meaning(semantics(event1(xvr)), simpleSemantics(xvr)),
  U)

```

At this stage, all the system knows is that this item خور / xvr is a verbal stem requiring an affix of type *1. Second, the system picks up the prefix مى / my. If we check the entry for مى / my we will see the following:

```

"my" $$ X :-
  affix@X <-> *1,
  X <> [verb, prefix, pres_tense],
  affixes@X <-> [AGR],
  subject@X <> nom,
  dir@AGR <> xafter,
  affix@AGR <-> *agr,
  [syntax, lex_type]@AGR <-> [syntax, lex_type]@X.

```

Now, the following information is provided to the system using the above entry:

The prefix مى / my is an affix of type *1;
 This affix is a verbal prefix and shows the present tense;
 The stem and the affix still need other affixes in the list to be saturated;
 There is only one item in the list, namely the [AGR]affix; this affix is a nominative affix which attaches to the right of the stem and should be of type *agr

Now, it is time to see what the system knows about مى / my and then the combination of the two:

```

sign(structure(direction(-after, +before), A),
  morph(affixes(["VP"5(B)], affix>(*1)), C),
  syn(nonfoot(head(cat(xbar(+v, -n)),
    vform(vfeatures(finite(+tensed,
      -participle,
      infinitive(-, D)),
      +active,
      view(tense(+present,
        -past,
        -future,
        -preterite,
        -free),
      aspect(simple)),
      E),
    subject(??? (F)),
    -gerund),
  G),

```

minor(target(H) mod I, -comp, J)),
 K),
 L)

These are the pieces of information provided to us about می / my by the system:

It is a *1 affix;
 It is marked as being present tense and active;
 It still needs another verbal affix

Now, time to see the system's output when it tries to parse, می خور / myxvr:

```
sign(morph(affixes(["VERB"(D)]),
  -affix,
  history(-umlauted, branch([], E),
  verb(d)),
  syn(nonfoot(head(cat(xbar(+v, -n)),
    vform(vfeatures(finite(+tensed,
      -participle,
      infinitive(-, F)),
      -aux,
      +active,
      view(tense(+present,
        -past,
        -future,
        -preterite,
        -free),
        aspect(simple)),
        mood(irreal(G), main(H))),
        subject("NOUN"(I)),
        -gerund),
    J),
  mcopy(type(K), L),
  minor(target(M) mod N,
  intensified(O),
  -conj,
  specf(kspec(+specified, P),
  specifier(time([(quant(existential, tense),
    ,(tense(+present,
      -past,
      -future,
      -preterite,
      -free),
      G))),
    aspect(simple)])),
  Q),
  -comp,
  R)),
  subcat(args(["NOUN"(_710,)), "NOUN"(I)], fixed(S)),
  foot(wh([], topicalised([]))),
  meaning(semantic(event1(xvr)), simpleSemantics(xvr)),
  remarks(failures(T), U))
```

At this stage, the system has combined the two elements, namely the stem, and the prefix. The output shows the system has acquired a large amount of information; however, in order for the verb to be fully saturated morphologically, it still requires the affix of type *agr as defined in the entry for می / my.

Now, let us have a look at the entry for the agreement marker


```

“m” $$ X :-
  affix@X <-> *agr,
  X <> [verb, suffix, first_sing_only].

```

The entry mentions that this affix is of type *agr, exactly what we need; however, in the entry for this affix, there is no mentioning of other affixes, and now the verb is fully saturated. Now we can see the output for the string می خورم / myxvrm:

```

sign(morph(affixes([]),
  -affix,
  history(-umlauted, branch([], D),
  verb(d)),
  syn(nonfoot(head(cat(xbar(+v, -n)),
    agree(first(+sing, -dual, -plural),
    second(-sing, -dual, -plural),
    third(-sing, -dual, -plural),
    count(+individual, -kind, -mass, E),
    F),
    vform(vfeatures(finite(+tensed,
      -participle,
      infinitive(-, G)),
      -aux,
      +active,
      view(tense(+present,
        -past,
        -future,
        -preterite,
        -free),
        aspect(simple)),
        mood(irreal(H), main(I))),
        subject("NOUN"(J)),
        -gerund),
    K),
  mcopy(type(L), M),
  minor(target(N) mod O,
  intensified(P),
  -conj,
  specf(kspec(+specified, Q),
    specifier(time([(quant(existential, tense),
      ,(tense(+present,
        -past,
        -future,
        -preterite,
        -free),
        H))),
    aspect(simple)])),
    R),
  -comp,
  S)),
  subcat(args(["NOUN"(_624,)), "NOUN"(J)], fixed(T)),
  foot(wh([]), topicalised([]))),
  meaning(semantics(event1(xvr)), simpleSemantics(xvr)),
  remarks(failures(U), V))

```

Notice the large amount of information gathered and displayed by the system via combining the three elements. This was a typical example, however, all morphological processes in Persian can and will be captured via the same mechanism. The information presented here is only general and due to limitations, we do not go more into the details. Let us bring this

discussion to an end by showing this process which is used by the system for some other morphological constructions in Persian using the following diagrams:

$$\left(\begin{array}{cc} (*1 & stem \\ my & xvr \end{array} \right) * AGR \\ m$$

Figure 1. Simple Present

The stem is combined with the *1 affix (in this case می / my) and then combines with the agreement marker to form a fully saturated lexical item.

$$\left(\begin{array}{cc} (*1 & stem \\ b & xvr \end{array} \right) * AGR \\ m$$

Figure 2. Subjunctive

ب / b is the imperative marker

$$\left(\begin{array}{c} *aspect \left(\begin{array}{cc} stem & *1 \\ xvr & d \end{array} \right) \\ my \end{array} \right) * AGR \\ m$$

Figure 3. Past Progressive

$$\left(\begin{array}{cc} stem & *1 \\ xvr & d \end{array} \right) * AGR \\ m$$

Figure 4. Simple Past

6 CONCLUSION

This paper was devoted to both theoretical discussion as well as computational description of Persian morphology. It is clear that by providing morphological analysis, the number of items in the dictionary file is drastically reduced. The approach to Persian Morphology in this paper has been the categorial one which has never been applied to Persian before and it is quite clear that most of the morphological features of the language have been captured. However, in order for the current system to be powerful, it should be tested with different data so that the weaknesses of the system are determined and steps are taken to account for these weaknesses.

References

- [1] E. L. Antworth, . *PC-KIMMO: A Two-Level Processor for Morphological Analysis*. Dallas, TX, Summer Institute of Linguistics. 1990
- [2] L.. Bauer, *English Word Formation*. Cambridge, CUP. 1983
- [3] Bennet,. *Feature-Based Approaches to Grammar*. Manchester, Language Engineering, UMIST. 1997
- [4] M.Dabir-Moghaddam, "*Morphological Causatives in Persian*." The 11th Annual Meeting of the Kansai Linguistics Society of Japan, Osaka, Japan. 1987a.
- [5] M. Dabir-Moghaddam,. "*Causative Constructions in Persian*." Iranian Journal of Linguistics 5(1): 13-76. 1987b
- [6] Dabir-Moghaddam, M.. "*Piramune 'ra' dar Zabane Farsi* [On ra in Persian Language]." *Majalleye Zabanshenasi* 7(1): 2-60. 1989
- [7] K. A. Farid, *A Discourse-Pragmatic Description of Marked Constructions in Persian*. PhD Thesis. Department of Language Engineering. Manchester, UMIST: 272. 1997.
- [8] A. Hajati,. "*Fe'le Lazem va ra dar Zabane Farsi* [Intransitive Verb and Ra in Persian Language]." *Majalleye Daneshkadeye Adabiyat va Olume Ensaniye Tarbiyat Mo'allem* 5: 185-211. 1976
- [9] M. R. Hashemi,. *A Contrastive Study of English and Persian Tense and Aspect Systems with Reference to Translation Practice*. PhD Thesis, Department of Language Engineering. Manchester, UMIST: 313. 1997
- [10] Hoeksma, J.. *Categorial Morphology*. New York, Garland Publishing. 1985

From Data to Nuanced Information: Making Implicit Knowledge Useful

Mona Diab

Columbia University

md2370@columbia.edu

*Abstract-*Natural Language processing (NLP) is a field concerned with the automatic processing of natural languages as they occur in the different communication media, spoken and written. NLP is especially important for converting raw data as it occurs in unstructured forms into information. With the advent of advanced statistical methods and machine learning techniques, we have witnessed a surge in the NLP technology reaching levels of unprecedented and even unexpected success in processing language. A lot of this success can be attributed to progress in the infrastructure machinery but also the sophisticated statistical methods employed. I hope I will be able to convince you that nuanced knowledge about the underlying data is crucial to break the current plateaus achieved. The devil is in the details. NLP is at the interface of multiple complex disciplines and in order to garner the next leap, there is a serious need for attention to detail; we should not only be concerned with what to model, but also how to model it. In this talk, I will discuss several information extraction problems. The problem of identifying who did what to whom, using semantic knowledge in the process of semantic role labeling (SRL); Is the speaker a person or geo political entity, can we tell the entity class of the White House when it issues a statement or when it is painted green, the problem of Named Entity Recognition. I will illustrate that different languages require different approaches that go beyond feature engineering. I will show you some examples of how important it is to pay attention to such nuanced information in the context of Question Answering (relevant for information retrieval) and Machine Translation.

Biography- Mona Diab is Research Scientist at the Center for Computational Learning Systems (CCLS) and an Adjunct Associate Professor in the Computer Science Department, at Columbia University. She is the co-founder of the CADIM (Columbia Arabic Dialect Modeling) research group. CADIM research focuses on addressing challenges in Arabic language processing taking into consideration different modalities and genres of data. Mona's areas of expertise include Multilingual processing, Computational lexical semantics, Information Extraction, Machine Translation, Computational Sociolinguistics, and last but not least, exploiting advanced Machine learning approaches for dialect processing. As extra-curricular activities, Mona currently serves as a re-elected Columbia University Senator representing the Columbia research community; she is also heavily engaged in promoting science and technology education for women and minorities, within Columbia and in the scientific community at large. Mona serves on several academic editorial boards. She is an elected secretary for the Association for Computational Linguistics Special Interest Group on Lexical Semantics (ACL SIGLEX). She also serves as the elected secretary for the ACL SIG on Semitic Language Processing. Mona has published over 70 publications in top tier conferences and scientific venues. Mona earned her PhD in Computational Linguistics in the University of Maryland in 2003 and went on to do postdoctoral research with Daniel Jurafsky at Stanford University, from 2003-2005.

Generating Lexical Resources for Opinion Mining in Arabic Language Automatically

Hanaa Bayomi Ali ^{*1}, Mohsen Rashwan ^{**2}, Samir Abd_Elrahman ^{*3}

**Computer Science Department, Faculty of Computers and Information and Cairo University
Giza 12613, Egypt*

¹*h.mobarz@fci-cu.edu.eg*

³*s.abdelrahman@fci-cu.edu.eg*

*** Electronics and Communications Department, Faculty of Engineering and Cairo University
The Engineering Company for the Development of Computer Systems; RDI, Al-Haram Av., 12111, Giza, Egypt*

²*Mohsen_Rashwan@RDI-eg.com*

Abstract— In this work we present SENTIRDI, a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications. We confront the task of deciding whether a given Arabic term has a positive connotation, or a negative connotation, or has no subjective connotation at all; this problem thus subsumes the problem of determining subjectivity and the problem of determining orientation. We tackle this problem by bootstrapping from three small sets of terms (Positive, Objective, and Negative seed sets) and increase sets consequently by applying lexical relation that is available in RDI Lexical Semantic Data Base (RDILSDB) until cover all Arabic semantic fields.

1 INTRODUCTION

Opinion mining is a recent sub discipline of computational linguistics which is concerned not with the topic a document is about, but with the opinion it expresses. Opinion-driven content management has several important applications, such as determining critics' opinions about a given product by classifying online product reviews, or tracking the shifting attitudes of the general public toward a political candidate by mining online forums.

Within opinion mining process several tasks are defined; these tasks involve tagging a given document depending on the opinion it express. The defined tasks are:-

- **Determining document subjectivity**, as in deciding whether a given text has a factual nature (i.e. describes a given situation or event, without expressing a positive or a negative opinion on it) or expresses an opinion on its subject matter. This amounts to performing binary text categorization under categories Objective and Subjective (Pang and Lee,2004; Yu and Hatzivassiloglou, 2003); ([1]; [2]);
- **Determining document orientation (or polarity)**, as in deciding if a given Subjective text expresses a positive or a negative opinion on its subject matter (Pang and Lee, 2004;Turney, 2002); ([1]; [3]);
- **Determining the strength of document orientation**, as in deciding whether the positive opinion expressed by a text is weakly positive, mildly positive, or strongly positive (Wilson et al., 2004) ([4]).

In order to aid these tasks, we need to identify the orientation of subjective terms contained in text, i.e. determining whether a term that carries opinionated content has a positive or a negative connotation.

Opinion Mining for Arabic language is considered a hot research topic with a very few contributions. This is due to the complexity of Arabic language and rareness of Opinion Mining Arabic Linguistic resources. This problem thus subsumes the problem of determining subjectivity and the problem of determining orientation.

One of the big challenges while dealing with term orientation in Arabic language for which one would like to perform opinion mining is that, there is no available lexical resource where terms are tagged as having either positive or negative connotation. The absence of such a resource emerged the need to generate it automatically.

2 RELATED WORK

A. Determining term orientation

Most previous work dealing with the properties of terms within an opinion mining perspective focused on determining term orientation.

Hatzivassiloglou and McKeown (1997) [5] attempt to predict the orientation of subjective adjectives by analysing pairs of adjectives (conjoined by ‘and’, ‘or’, ‘but’, ‘either-or’, or ‘neither-nor’) extracted from a large unlabelled document set. The underlying intuition is that the act of conjoining adjectives is subject to linguistic constraints on the orientation of the adjectives involved; e.g. ‘and’ usually conjoins adjectives of equal orientation, while ‘but’ conjoins adjectives of opposite orientation. The authors generate a graph where terms are nodes connected by “equal-orientation” or “opposite-orientation” edges, depending on the conjunctions extracted from the document set. A clustering algorithm then partitions the graph into a Positive cluster and a Negative cluster, based on a relation of similarity.

Turney and Littman (2003) [6] determine term orientation by bootstrapping from two small sets of subjective “seed” terms (with the seed set for Positive containing terms such as good and nice, and the seed set for Negative containing terms such as bad and nasty). Their method is based on computing the point wise mutual information (PMI) of the target term t with each seed term t_i as a measure of their semantic association. Given a target term t , its orientation value $O(t)$ (where positive value means positive Orientation, and higher absolute value means stronger orientation) is given by the sum of the weights of its semantic association with the seed positive terms minus the sum of the weights of its semantic association with the seed negative terms. For computing PMI, term frequencies and co-occurrence frequencies are measured by querying a document set by means of the AltaVista search engine with a “ t ” query, a “ t_i ” query, and a “ t NEAR t_i ” query, and using the number of matching documents returned by the search engine as estimates of the probabilities needed for the computation of PMI.

Kamps et al. (2004) [7] consider instead the graph defined on adjectives by the WordNet 2 synonymy relation, and determine the orientation of a target adjective t contained in the graph by comparing the lengths of (i) the shortest path between t and the seed term good, and (ii) the shortest path between t and the seed term bad: if the former is shorter than the latter, than t is deemed to be Positive, otherwise it is deemed to be Negative.

Takamura et al. (2005) [8] determines term orientation (for Japanese) according to a “spin model”, i.e. a physical model of a set of electrons each endowed with one between two possible spin directions, and where electrons propagate their spin direction to neighbouring electrons until the system reaches a stable configuration. The authors equate terms with electrons and term orientation to spin direction. They build a neighbourhood matrix connecting each pair of terms if one appears in the gloss of the other, and iteratively apply the spin model on the matrix until a “minimum energy” configuration is reached. The orientation assigned to a term then corresponds to the spin direction assigned to electrons

The system of Kim and Hovy (2004) [9] tackled orientation detection by attributing, to each term, a positivity score and a negativity score; interestingly, terms may thus be deemed to have both positive and negative correlation, maybe with different degrees, and some terms may be deemed to carry a stronger positive (or negative) orientation than others. Their system starts from a set of positive and negative seed terms, and expands the positive (resp. negative) seed set by adding to it the synonyms of positive (resp. negative) seed terms and the antonyms of negative (resp. positive) seed terms. The system classifies then a target term t into either positive or negative by means of two alternative learning-free methods based on the probabilities that synonyms of t also appear in the respective expanded seed sets. A problem with this method is that it can classify only terms that share some synonyms with the expanded seed sets.

The method of (Esuli and Sebastiani, 2005) [10] starts from two small seed (i.e. training) sets L_p and L_n of known positive and negative terms, respectively, and expands them into the two final training sets $Trp \supset L_p$ and $Trn \supset L_n$ by adding them new sets of terms up and unfound by navigating the WordNet graph along the synonymy and antonymy relations. This process is based on the hypothesis that synonymy and antonymy, in addition to defining a relation of meaning, also define a relation of orientation, i.e. that two synonyms typically have the same orientation and two antonyms typically have opposite orientation.

When tested on the same benchmarks, the methods of (Esuli and Sebastiani, 2005; Turney and Littman, 2003) [10,6] performed with comparable accuracies (however, the method of (Esuli and Sebastiani, 2005) [10] is much more efficient than the one proposed by (Turney and Littman, 2003) [6]), and have outperformed the method of (Hatzivas-siloglou and McKeown, 1997) [5] by a wide margin and the one by (Kamps et al., 2004) [7] by a very wide margin. The methods described in (Hatzivassiloglou and McKeown, 1997) [5] is also limited by the fact that it can only decide the orientation of adjectives, while the method of (Kamps et al., 2004) [7] is further limited in that it can only work on adjectives that are present in WorldNet. The methods of (Kim and Hovy, 2004; Takamura et al., 2005) [9,8] are difficult to be compared with the other methods since they were not evaluated on publicly available datasets.

B. Determining term subjectivity

Riloff et al. (2003) [11] developed a method to determine whether a term has a subjective or an objective connotation, based on bootstrapping algorithms. The method identifies patterns for the extraction of subjective nouns from text, bootstrapping from a seed set of 20 terms that the authors judge to be strongly subjective and have found to have high frequency in the text collection from which the subjective nouns must be extracted. The results of this method are not easy to compare with the ones we present in this paper because of the different evaluation methodologies. While we adopt the evaluation methodology used in all of the papers reviewed so far (i.e. checking how good our system is at replicating an existing, independently motivated lexical resource), the authors do not test their method on an independently identified set of labelled terms, but on the set of terms that the algorithm itself extracts. This evaluation methodology only allows testing precision, and not accuracy tout court, since no quantification can be made of false negatives (i.e. the subjective terms that the algorithm should have spotted but has not spotted). This will prevent us from drawing comparisons between this method and our own.

Baroni and Vegnaduzzo (2004) [12] apply the PMI method first used by Turney and Littman (2003) [6] to determine term orientation and subjectivity. Their method uses a small set S_s of 35 adjectives, marked as subjective by human judges, to assign a subjectivity score to each adjective to be classified. Therefore, their method, unlike our own, does not classify terms (i.e. take firm classification decisions), but ranks them according to a subjectivity score, on which they evaluate precision at various level of recall.

C. Multilingual Sentiment Analysis

There is a growing body of work on multilingual sentiment analysis. Most approaches focus on resource adaptation from one language (usually English) to another with few sentiment resources. Mihalcea et al. (2007)[13], for example, generate subjectivity analysis resources in a new language from the English sentiment resources by leveraging a bilingual dictionary or a parallel corpus. Banea et al. (2008; 2010) [14,15] instead automatically translate the English resources by using automatic machine translation engines for subjectivity classification. Prettenhofer and Stein (2010)[16] investigate cross-lingual sentiment classification from the perspective of domain adaptation based on structural correspondence learning (Blitzer et al., 2006)[17]. Approaches that do not explicitly involve resource adaptation include (Wan (2009))[18], which uses co-training (Blum and Mitchell, 1998)[19] with English vs. Chinese features comprising the two independent "views" to exploit unlabeled Chinese data and a labeled English corpus and thereby improves Chinese sentiment classification.

Another notable approach is the work of Boyd-Graber and Resnik (2010)[20], in which they present a generative model, supervised multilingual latent Dirichlet allocation, by jointly modeling topics that are consistent across languages, and employing them to better predict sentiment ratings.

Unlike the methods described above, we focus on simultaneously improving the performance of sentiment classification in a pair of languages by developing a model that relies on sentiment-labelled data in each language as well as unlabelled parallel text for the language pair.

D. SentiWordNet

SentiWordNet, a lexical resource produced by asking an automated classifier $\hat{\Phi}$ to associate to the unique sense represented by each synset s of WordNet (version 2.0) a triplet of scores $\hat{\Phi}(s, p)$ (for $p \in P = \{\text{Positive, Negative, Objective}\}$) describing how strongly that sense enjoy each of the three properties. The method used to develop SentiWordNet is based on the term classification. The score triplet is derived by combining the results produced by a committee of eight ternary classifiers, all characterized by similar accuracy levels but extremely different classification behaviours. (Esuli and Sebastiani, 2005)[10]

3 DETERMINING SUBJECTIVITY & ORIENTATION OF ARABIC TERMS

We present a method for determining term orientation and term subjectivity using semi-supervised technique. Our process is composed of the following steps:-

1. Seed sets (S_p, S_n, S_o) represent three seed sets one for a positive set, one for a negative set and the last for objective set. They are provided as input.
2. Apply lexical Relation (Causality (القحط و الإهلاك), (Causative (شدة الحُب و الألفة), antonym (الحب و الكره), hyponymy (الخبانة و الذنب) and Hypernym (e. g. الجريمة و الخيانة)) from a Lexical semantic Data Base for each term in (S_p, S_n), in order to find new semantic fields and increase the size of seed set. The new semantic fields, once added to the original ones, yield two new, richer sets \hat{S}_p and \hat{S}_n of semantic fields.

The method requires bootstrapping from a seed sets (Sp, Sn, So) representative of the categories Positive, Negative and Objective. In our experiments we begin from Turney and Littman (2003), (Esuli and Sebastiani, 2005) ([10],[6])seed sets. In order to classify the largest number of semantic fields in Semantic Data Base we increase the number of terms gradually in seed sets by noticing the results until we reach the satisfied result.

For objective seed set, we should notice that in previous work objective terms can be concluded from positive and negative terms [10] but here we begin from seed set in order to improve the results.

B. Expansion method for seed sets

We use RDI Lexical Semantic Data Base (RDILSDB) as the source of lexical relations. (RDILSDB) contains 18.413 semantic fields. It covers 100.000 words. Semantic fields relate together with 293,000 Bilateral semantic via 20 lexical relations (Part_of, Totality, Inclusion_K, KindOf, Inclusion_M, Member_of, Inclusion_I, Integeration, Inclusion_O, Original, Conditional, Required_Condition, Causality, Causative, Circumstantial_Place, Locality_Place, Circumstantial_Time, Locality_Time, synonym and Antonymy).

5 RESULTS

In fact, fully testing the accuracy of our tagging method experimentally is impossible, since this would require a version of all semantic fields in Arabic language manually annotated according to our three properties of interest, and the unavailability of such a manually annotated resource is exactly the reason why we are interested in generating it automatically.

Our proposed work is evaluated by two methods:-

The first method is using a manually annotated subset of Arabic semantic fields as a “gold standard”. it is annotated by 5 annotator (3 Linguistics from an engineering company to develop digital system (RDI) and 2 Linguistics from Faculty of Dar Science, Cairo university)

Number of Semantic Fields in a “gold standard” is 7216. Table1 represent recall and precision of positive, negative, objective and all semantic fields' regardless polarity after applying proposed algorithm.

The second method is Translating The Micro-WNOp[S. Cerini, V. Compagnoni, A. Demontis, M 2007] (by Google translator) gold standard that is used to evaluate the Senti wordnet which contains 1.105 synset. Table 2 represents recall and precision of positive, negative, objective and all semantic fields' regardless polarity after applying proposed algorithm.

TABLE1

REPRESENT RECALL AND PRECISION OF POSITIVE, NEGATIVE, OBJECTIVE AND ALL SEMANTIC FIELDS FOR THE FIRST TEST.

	Pos_SF	Neg_SF	Obj_SF	All_SF
Recall	86.44	83.16	89.3	87.32
Precision	74.76	79.95	93.09	87.72
F-measure	80.18	81.52	91.16	87.52

TABLE2

REPRESENT RECALL AND PRECISION OF POSITIVE, NEGATIVE, OBJECTIVE AND ALL SEMANTIC FIELDS FOR THE SECOND TEST.

	Pos_SF	Neg_SF	Obj_SF	All_SF
Recall	79,43	85	89	84.67
Precision	80	82,45	87.93	84.95
F-measure	79,71	83.7	88.46	84.81

6 CONCLUSIONS

We have presented a method for determining both term subjectivity and term orientation for using semi-supervised technique which can be considered a very useful tool for all Arabic opinion mining applications because of its wide coverage (all LSDB Semantic fields are tagged according to each of the three labels Objective, Positive and Negative).

It is found that the direct translation from one language (English) to other language (Arabic) doesn't give accurate results because the same term may have a lot of meaning with different polarity. (Ex. Ball has the following meaning (لعبة من ألعاب الكرة, جسم مستدير من الإنسان, رصاصة, نزهة, حفلة راقصة, كرة)

After a lot of experiments it is found that (RDILSDB) doesn't recognize on countries nouns, numbers and Currency On the contrary Wordnet does.

ACKNOWLEDGMENT

This work was partially supported by Project ICT Centers of Excellence (CoE) "Data Mining and Computer Modeling", funded by Itida.

REFERENCES

- [1] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL-04, 42nd Meeting of the Association for Computational Linguistics*, pages 271–278, Barcelona, ES.
- [2] Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP-03, 8th Conference on Empirical Methods in Natural Language Processing*, pages 129–136, Sapporo, JP.
- [3] Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-02, 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, US.
- [4] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of AAAI-04, 21st Conference of the American Association for Artificial Intelligence*, pages 761–769, San Jose, US.
- [5] Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181, Madrid, ES.
- [6] Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.
- [7] Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation*, volume IV, pages 1115–1118, Lisbon, PT.
- [8] Hiroya Takamura, Takashi Inui, and Manabu Okumura 2005. Extracting emotional polarity of words using spin model. In *Proceedings of ACL-05, 43rd Annual Meeting of the Association for Computational Linguistics*, pages 133–140, Ann Arbor, US.
- [9] Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING-04, 20th International Conference on Computational Linguistics*, pages 1367–1373, Geneva, CH.
- [10] Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss analysis. In *Proceedings of CIKM-05, 14th ACM International Conference on Information and Knowledge Management*, pages 617–624, Bremen, DE.
- [11] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of CONLL-03, 7th Conference on Natural Language Learning*, pages 25–32, Edmonton, CA.
- [12] M. Baroni and S. Vegnaduzzo. 2004. Identifying subjective adjectives through Web-based mutual information. In *Proceedings of KONVENS-04, 7th Konferenz zur Verarbeitung Natürlicher Sprache (German Conference on Natural Language Processing)*, pages 17–24,

	<i>Vienna, AU.</i>
[13]	<i>Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In Proceedings of ACL'07.</i>
[14]	<i>Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2010. Multilingual subjectivity: Are more languages better? In Proceedings of COLING'10.</i>
[15]	<i>Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. In Proceedings of EMNLP'08.</i>
[16]	<i>Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In Proceedings of ACL'10.</i>
[17]	<i>John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In Proceedings of EMNLP'06.</i>
[18]	<i>Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In Proceedings of ACL/AFNLP'09.</i>
[19]	<i>Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In Proceedings of COLT'98.</i>
[20]	<i>Jordan Boyd-Graber and Philip Resnik. 2010. Holistic sentiment analysis across languages: Multilingual supervised Latent Dirichlet Allocation. In Proceedings of EMNLP'10.</i>

Analyzing Arabic Diacritization Errors of MADA and Sakhr Diacritizer

Hamdy Mubarak, Ahmed Metwally, Mostafa Ramadan

Arabic NLP Researches, Sakhr Software
Sakhr Building, Free Zone, Nasr City 11711,
Cairo, Egypt
{hamdys, amt, msr}@sakhr.com

Abstract— Modern standard Arabic (MSA) is usually written without diacritics, and this leads to morphological, syntactic, and semantic ambiguity. Diacritization (or diacritic restoration) is a very important basic step for several natural language processing (NLP) applications. In this paper, we present Sakhr Arabic disambiguation system that is used for selecting the best diacritization and sense for all words in Arabic text. We compare with the best performing reported system of Habash and Rambow (MADA) by analyzing errors in stem diacritization and case ending diacritization (using random samples from the GALE Dev10 newswire development data). We report the word error rate (WER) and diacritic error rate (DER) for both systems. Also, we give detailed statistics about different kinds of diacritization errors.

1 INTRODUCTION

Arabic is written with an orthography that includes optional diacritics typically representing short vowels. The absence of diacritics in modern standard Arabic (MSA) text is one of the most critical problems facing computer processing of Arabic text since this adds another layer of morphological and lexical ambiguity (one written word form can have several pronunciations, each pronunciation carrying its own meaning(s)).

Diacritization (aka vowelization, diacritic/vowel restoration) of Arabic text helps clarify the meaning of words and disambiguate any vague spellings or pronunciations. Diacritization is an important processing step for several natural language processing (NLP) applications, including part of speech (POS) disambiguation, training language models for Automatic Speech Recognition (ASR), Text-To-Speech (TTS) generation (Habash and Rambow 2007), in addition to Machine Translation (MT), and Arabic Data Mining applications (Shaalán et al., 2009).

Naturally occurring Arabic text has some percentage of diacritics, depending on genre and domain, to aid the reader disambiguate the text or simply to articulate it correctly. For instance, religious text such as the Holy Quran is fully diacritized to minimize the chances of reciting it incorrectly. Children’s educational texts and classical poetry tend to be diacritized as well. However, news text and other genre are sparsely diacritized (e.g., around 1.5% of tokens in the United Nations Arabic corpus bear at least one diacritic) (Diab et al., 2007).

In this paper, we evaluate and analyze errors for two famous diacritization systems, namely the Morphological Analysis and Disambiguation of Arabic (MADA) system (Habash and Rambow, 2005) and Sakhr Arabic Disambiguation System (ADS). The purpose is to highlight the most common errors in diacritization systems that need more focus and analysis to enhance accuracy.

This paper is organized as follows: Section 2 gives some examples and statistics about ambiguity in Arabic text due to lack of diacritics. Section 3 gives an overview about MADA. Section 4 describes Sakhr ADS. As for Section 5, it presents two experiments for evaluating these diacritization systems and detailed error analysis for each. Finally, section 6 gives some concluding remarks.

2 AMBIGUITY OF ARABIC LANGUAGE

Arabic is a highly inflected language which has a rich and complex morphological system. MSA is very often written without diacritics, which leads to a highly ambiguous text. Arabic readers could differentiate between words having the same writing form (homographs) by the context of the script. For example, the word “علم”¹ can be diacritized as “عِلْمَ Eilm, science or knowing”, “عَالِمًا Ealima, knew”, “عَلَّمَ Eallama, taught”, “عَلَمَ Ealam, flag”, etc.

¹ We use Buckwalter Arabic transliteration (Buckwalter, 2002) (<http://www.qamus.org/transliteration.htm>).

Debili, et al. (2002) calculate that an Arabic non-diacritized dictionary word form had 2.9 possible diacritized forms on average, and that an Arabic text containing 23K word forms showed an average ratio of 1:11.6 (quoted in Vergyri & Kirchoff 2004) (Maamouri et al., 2006).

Maamouri and Bies (2010) show 21 different analyses of the Arabic word “ثمن” *vmn*, produced by BAMA. At SYSTRAN, which has been developing machine translation systems for over 40 years, it was estimated that the average number of ambiguities for a token in most languages was 2.3, whereas in MSA it reaches 19.2. Although ambiguity is caused primarily by the absence of short vowels, at SYSTRAN, researchers have found ambiguity in Arabic to be present at every level (Farghaly and Shaalan, 2009).

A. MSA Ambiguity in a POS-Tagged Corpus

For Sakhr POS-tagged corpus that contains 7M words gathered from different modern news services, we observed that MSA tends to be simpler than the Classical Arabic in grammar usage, syntax structure, morphological and semantic ambiguity. This helps normal Arabic readers to understand the written text easily. For example, 69% of words in this corpus have only 1 identified morphological analysis (one morphological interpretation), and 19% have 2 analyses, while high ambiguous words (3+ analyses) represent 12% only (Mubarak et al., 2009) as shown in Figure 1.

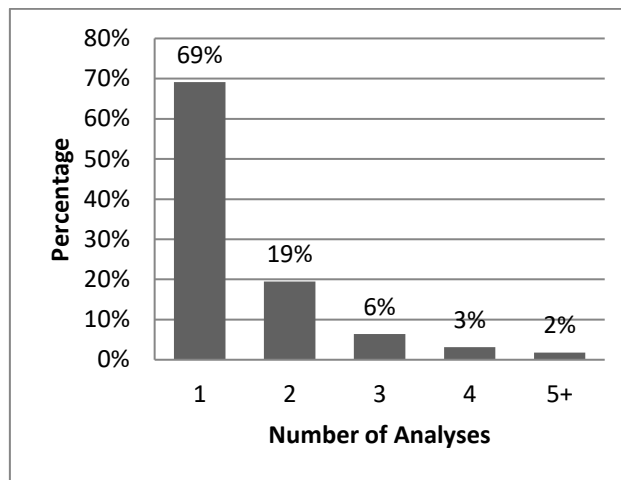


Figure 1: Distribution of Number of Word Analyses

Because Sakhr Morphological Analyzer provides an ordered list of analyses according to usage frequency, it was discovered that 92% of words occupy the first position in analyses, and 5% occupy the second one as shown in Figure 2, which means that MSA in most cases is not so ambiguous, and words occupy the “trivial” analysis! For example, the word “للحاكم” *lilHaAkim* has more than one analysis (للحاكم *liloHaAkimi*, to/of/for the ruler, للحاكم *liliHaAkumo*, to/of/for your beards, etc.), but the first one is usually recognized.

Figure 3 shows the distribution of case ending marks (mark on last letter) for nouns and verbs. We can observe that the case ending for verbs (if not given غير مبني *غير مبني*) tends to be indicative (~81% of the cases), and for nouns (if not given) it tends to be genitive (~56% of the cases).

Figure 4 shows the distribution of diacritics extracted from the fully diacritized corpus. It is notable that “Fatha” is the most frequent diacritic and forms with “Kasra”, “Sukun” and “Damma” represent ~97% of the whole diacritics.

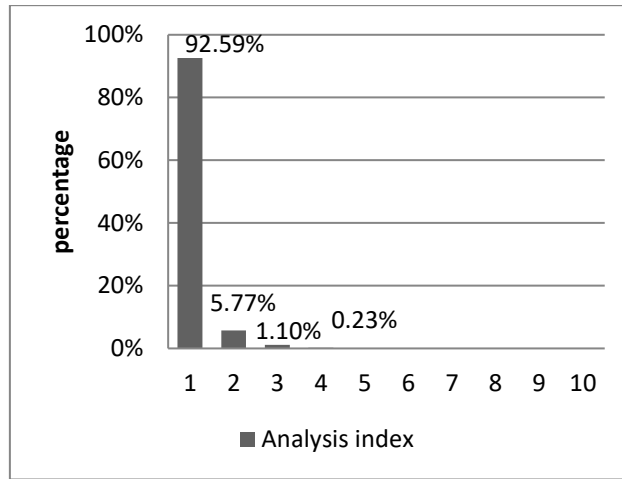


Figure 2: Distribution of the Selected Analysis Index

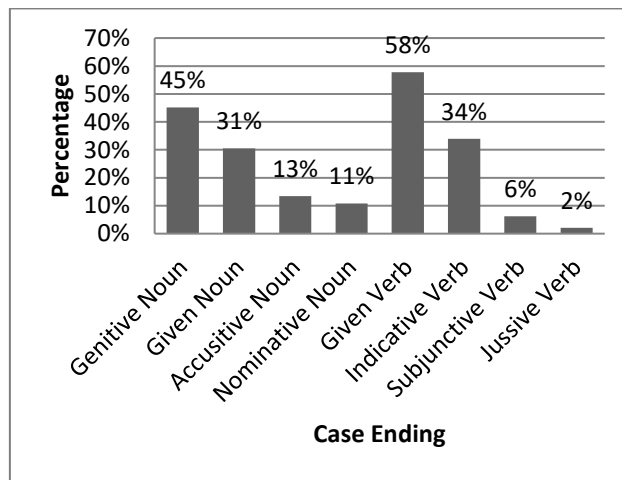


Figure 3: Case Ending Distribution

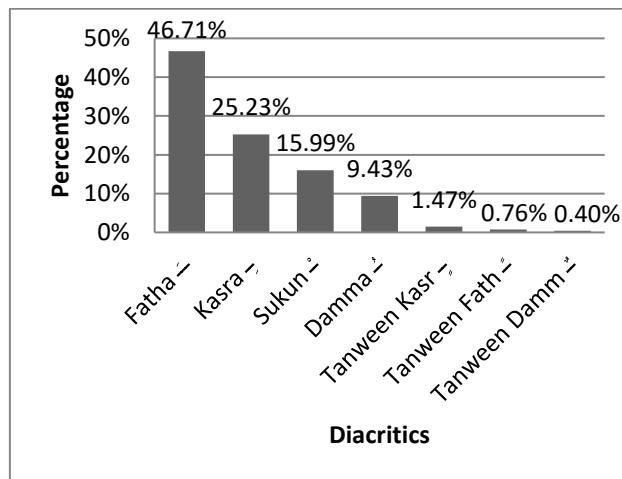


Figure 4: Diacritics Distribution

3 THE MADA SYSTEM

As mentioned in (Habash and Rambow, 2005), the basic approach used in MADA is inspired by the work of Hajic (2000) for tagging morphologically rich languages, which was extended to Arabic independently by Hajic et al. (2005). In this approach, a set of taggers are trained for individual linguistic features which are components of the full morphological tag (such as core

part-of-speech, tense, number, and so on). In Arabic, we have ca. 2,000 to 20,000 morphological tags, depending on how we count. The Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004) is consulted to produce a list of possible analyses for a word. BAMA returns, given an undiacritized inflected word form, all possible morphological analyses, including full diacritization for each analysis. The results of the individual taggers are used to choose among these possible analyses. The algorithm proposed for choosing the best BAMA analysis simply counts the number of predicted values for the set of linguistic features in each candidate analysis.

Habash and Rambow (2007) introduced a system called MADA-D that uses Buckwalter's Arabic morphological analyzer where they used 14 taggers and a lexeme-based language model.

4 SAKHR ARABIC DISAMBIGUATION SYSTEM (ADS)

Sakhr morphological analyzer is a morphological analyzer-synthesizer that provides basic analyses of a single Arabic word, covering the whole range of modern and classical Arabic. For each analysis, it provides its morphological data such as diacritization, stem, root, morphological pattern, POS, prefixes, suffixes and also its morphosyntactic features like gender, number, person, case ending, etc. In addition to its high accuracy (99.8%), the morphological analyzer sorts the word analyses according to the usage frequency (using manual ordering of analyses for commonly-used words as appeared in an Arabic corpus of 4G words, or ordering according to stem frequency, otherwise). This morphological analyzer is integrated in most Sakhr products like TTS, MT, Search Engine and Text Mining.

ADS selects the best morphological analysis (which carries a large set of morphological data), and the best sense (which carries a large set of semantic data). Figure 5 is a screen shot that shows the diacritization for a random sentence¹.

Figures 6-8 show the ADS morphological data (POS, diacritized stem, prefixes, suffixes, pattern, gender, number, person, etc), syntactic data (case ending, and attached pronoun), and semantic data (Arabic and English senses, semantic, ontological and thematic features).



Figure 5: ADS Diacritization

¹ ADS can be tested using website: <http://arabdiac.sakhr.com.eg>

Linguistic Data البيانات اللغوية - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://arabdiac.sakhr.com.eg/LinguisticData.aspx?Src=

Most Visited Getting Started Latest Headlines ---EGYTYRES--- Customize Links Egypt Software Online Dictionary at D...

Linguistic Data البيانات اللغوية

Arabic Text Diacritization

Trial Version 1.0

sakhr سحر www.sakhr.com

تشكيل النصوص العربية

نسخة تجريبية 1.0

Lexical Data بيانات معجمية Syntactic Data بيانات نحوية Morphological Data بيانات صرفية

بيانات صرفية							الكلمة						
نوع المفرد	الشمخص	العدد	النوع	التعريف	التمييز الصرفي	الجزر	قسم النظم	الثوابق	أصل الكلمة	المساويق	الكلمة المشكّلة	الكلمة	رقم
	مخاطب، غائب	مفرد	مذكر، مؤنث		يُفَعِّلُ	ص ر ر	فعل مضارع مزيد		وُصِرَ	و	وُصِرَ	وُصِرَ	1
		مفرد	مؤنث	معرفة			علم		إِزَان		إِزَان	إِزَان	2
		مفرد	مذكر	نكرة	فَاعِلٌ	د و م	صفة مشبّهة	أ	ذَالِمٌ		ذَالِمًا	ذَالِمًا	3
							حرف جر		عَلَى		عَلَى	عَلَى	4
		مفرد	مذكر	معرفة	فُعِلَ	ح ق ق	اسم ذات	هَآ	حَقٌّ		حَقَّهَا	حَقَّهَا	5
							حرف جر		فِي		فِي	فِي	6
		مفرد	مذكر	معرفة	فُعِلَ	ق و م	مصدر مجرد		فِيَامٌ	أَلْ	أَفِيَامٌ	أَفِيَامٌ	7
		مفرد	مؤنث	معرفة	فُعِلَ	ع م ل	اسم ذات		عَمَلِيَّةٌ	بِ	بِعَمَلِيَّةٍ	بِعَمَلِيَّةٍ	8
		مفرد	مذكر	معرفة	تَفَعَّلَ	خ ص ب	مصدر مزيد		تَخَصَّبَ		تَخَصَّبَ	تَخَصَّبَ	9
		مفرد	مذكر	معرفة			اسم		يُورَانِيُومٌ	أَلْ	أَلْيُورَانِيُومٌ	أَلْيُورَانِيُومٌ	10

Done

Figure 6: ADS Morphological Disambiguation

Linguistic Data البيانات اللغوية - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://arabdiac.sakhr.com.eg/LinguisticData.aspx?Src=

Most Visited Getting Started Latest Headlines ---EGYTYRES--- Customize Links Egypt Software Online Dictionary at D...

Linguistic Data البيانات اللغوية

Arabic Text Diacritization

Trial Version 1.0

sakhr سحر www.sakhr.com

تشكيل النصوص العربية

نسخة تجريبية 1.0

Lexical Data بيانات معجمية Syntactic Data بيانات نحوية Morphological Data بيانات صرفية

بيانات نحوية		الكلمة		
الضمير المنصل	الحالة الإعرابية	الكلمة المشكّلة	الكلمة	رقم
	مرفوع بالضمّة	وُصِرَ	وُصِرَ	1
	مرفوع	إِزَان	إِزَان	2
	منصوب بالفتحة	ذَالِمًا	ذَالِمًا	3
	مبني على السكون	عَلَى	عَلَى	4
ضمير جر	مجرور بالكسرة	حَقَّهَا	حَقَّهَا	5
	مبني على السكون	فِي	فِي	6
	مجرور بالكسرة	أَفِيَامٌ	أَفِيَامٌ	7
	مجرور بالكسرة	بِعَمَلِيَّةٍ	بِعَمَلِيَّةٍ	8
	مجرور بالكسرة	تَخَصَّبَ	تَخَصَّبَ	9
	مجرور	أَلْيُورَانِيُومٌ	أَلْيُورَانِيُومٌ	10

Done

Figure 7: ADS Syntactic Disambiguation

بيانات معجمية		الكلمة			
السمات الدلالية	المعنى بالإنجليزية	المعنى بالعربية	الكلمة المفصلة	الكلمة	رقم
أفعال القصد والتبوء، ثم وضوح أوحده، ثم شمول أو عموم، حقيقي، ليس له زمن محدد، مجرد، حيادية الحدث، ليس فعل حالة، ليس فعل إجراء، ليس فعل إنتاج، فعل إنجاز، فعل بلا نتيجة، فعل يحدث مرة واحدة، إرادة مختفية، إرادي، موضوع عام،	intენტness, insistence	على الأمر : ثبت عليه	وَتَصْرُ	وَتَصْرُ	1
بلا ومنظمة، له مساحة، له طول، له حجم الدولة، ليس له زوايا حادة، له حيز، ثابت لا يتحرك، ليس له سرعة، قطعة واحدة متجانسة، له جمال وفتح، له غمُر وكِدَم، له صحة وعلة، أرضي التواجد، له ذكاء وشجاعة، له قوة وضعف، له كدرة النفسية، له كدرة الإبصار، له كدرة السمع، له كدرة التمس، له خصائل ومميزات، له مكانة اجتماعية، له صفات دينية، له سنطة المثلث، ليس له سنطة، رعايا الدولة، له أهمية أو ثقاهة، حقيقي، لا يحتوي طاقة كامنة، مادي، صناعي، مفتوح، علم على آخر، في مستوى الكل، حيادية الحدث، جغرافي،	Iran	جمهورية إسلامية هي آسيا	إيران	إيران	2

Figure 8: ADS Semantic Disambiguation

The ADS block diagram shown in Figure 9 describes the basic components and processing steps to disambiguate Arabic texts. Processing starts by **segmenting** Arabic text into sentences taking into consideration CR/LF (Enter) characters, and the ambiguity in dots (end of sentence, or part of abbreviations or proper nouns). **Tokenization** step splits text into logical units (or tokens) considering special cases for punctuations, digits, abbreviations, URLs, etc. The **morphological analyzer** and **lexicalizer** provide different alternatives (analyses) for all words, and a large set of morphological, syntactic, and semantic information (including ontological features and attributes).

The **proper** database (~300K entries) is used to detect different types of named entities like: human, location, organization, etc. **Spelling correction** engine is then used to detect and correct offline errors (~1M entries) and online errors. Idioms, adverbs, and conjunctions are detected using the **idiom parser** which handles a database of basic forms (~100K entries) and their morphological expansions. Heuristics rules for function words are applied in the **Prelex** engine. **Collocates** and frequently used expressions (~3M entries) are handled using the collocations detector for continuous and non continuous words.

A statistical **POS-Tagger** is then used to select the best analysis (based on a POS-tagged corpus of 7M words).

Surface rules are then applied for special behaviors of words (like preposition attachment, and syntactic behaviors for “Haal الحال” and “التمييز Tamyeez”). For POS, case ending, and sense disambiguation, thousands of **grammar rules** are used to select the best solution. For example, a rule for detecting a DATE looks like¹:

DATE→*DAY *NUM *MONTH *NUM *H/M (to detect الجمعة 10 رمضان 1410 هـ، الثلاثاء 25 يناير 2011 م), and a rule to detect NUMBER looks like:

NUM→*NUM3:10 *NUM000 *N (to detect عشرة ملايين دينار، آلاف رجل 3 etc).

Theme disambiguation engine is finally used to resolve any residual ambiguity that can be solved using sentence dominant theme.

¹ The morphological analyzer provides these notations (pre-terminals) as part of syntactic data for all senses.

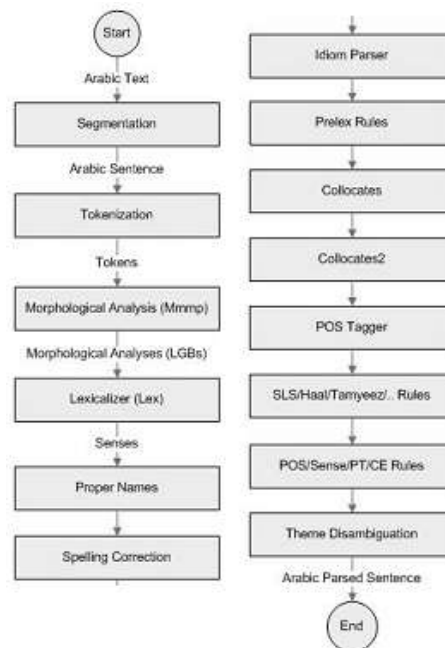


Figure 9: ADS Block Diagram

5 ANALYZING DIACRITIZATION ERRORS

Diacritization errors are usually calculated using two error rates: word error rate (WER) which indicates how many words have at least one diacritic error, and diacritic error rate (DER) which indicates how many letters we have incorrectly restored their diacritics.

Habash and Rambow (2007) mentioned that MADA is so far the best performing system to date. It has been reported that it achieved a WER of 14.9% and a DER of 4.8% compared with that of (Zitouni et al., 2006) which gives WER of 18.0% and DER of 5.5%.

It is worth mentioning that Shaalan et al., (2009) presented a hybrid approach for building Arabic diacritizer that gets results comparable with MADA with a WER of 11.8% and a DER of 3.2%.

Also, Rashwan et al., (2011) introduced a stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features. They compared their system with of Habash and Rambow, and of Zitouni, using the same training and test corpus for the sake of fair comparison. The word error rates of (morphological diacritization, overall diacritization including the case endings) for the three systems are, respectively, as follows (3.1%, 12.5%), (5.5%, 14.9%), and (7.9%, 18%).

We extracted 2 samples (each sample contains 100 sentences or ~10,000 words) from the GALE DEV10 Newswire set (1089 sentences) under the DARPA GALE program1. These samples are diacritized using MADA2 and Sakhr ADS.

We calculated errors manually for MADA and ADS considering **stem diacritization** (تشكيل البنية) and **case ending diacritization** (تشكيل الإعراب) for both samples³. We differentiate here between these errors as we believe that errors in stem diacritization are more important than errors in case ending diacritization for wide range of applications like TTS, MT, and text mining because this affects word meaning in most cases.

We found that number of stem diacritization errors for both samples for MADA was 141 (which represents 1.3%), and 108 (1.06%), while for ADS, the number was 35 (0.05%), and 32 (0.3%), and number of case ending diacritization errors for MADA was 509 (4.7%), and 400 (3.93%), while for ADS, the number was 222 (2.0%), and 180 (1.76%). Figure 10 shows these results.

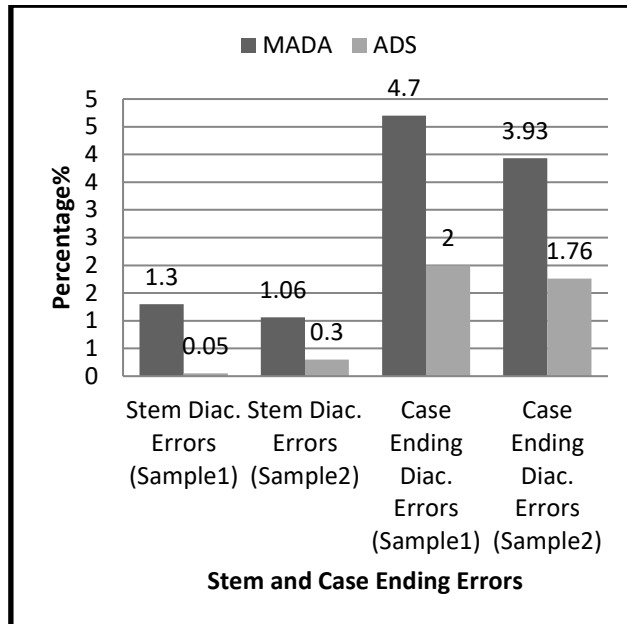


Figure 10: Stem and Case Ending Errors for MADA & ADS

A. Analyzing Stem Diacritization Errors

Error analysis for MADA shows that, on the average, 34% of stem diacritization errors are due to the lack of diacritics for unknown proper names, 30% are due to selecting wrong POS, and 16% are due to diacritizing some particles and function words incorrectly (namely, >n أن, <n إن, and mn من). The rest of errors (~20%) are mainly related to spelling mistakes and out of vocabulary (OOV) words. Figure 11 shows these errors in details and table I lists some examples for each type of errors.

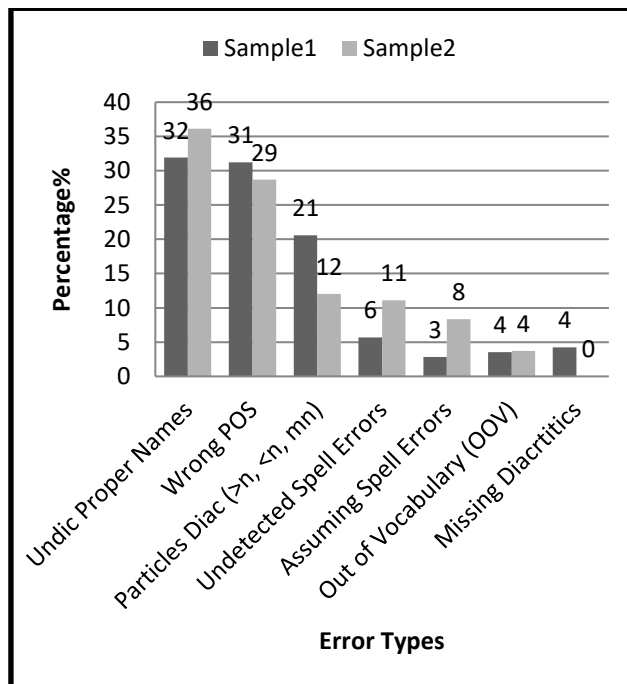


Figure 11: Error Analysis of Stem Diac. for MADA

¹ <http://www ldc.upenn.edu/>

² We thank Nizar Habash for sharing MADA's output

³ If a word has any error in its stem diacritization, we count this as stem error, and if a word has any error in its case ending diacritization only, we count this as case ending error.

TABLE I
ANALYSIS of STEM DIAC. ERRORS for MADA

MADA	
أمثلة	الخطأ
وَقَدْ أَلْفَى سِرْكَيْسِيَانُ أَمْسِ كَاتُولِيكُوسَ الْأَرْمَنَ / وَالنَّائِبُ أَعُوبُ بَقْرَادُونِيَانُ / تَقْرِيرُ غُولْدِسْتُونِ	أعلام غير مشكلة
الإِتِّفَاقُ التَّرْكِييُّ الْأَرْمَنِيُّ الْمُزْمَعُ عَقْدُهُ / وَأَبْرَزُ قِيَادِييِّهَا فِي قِطَاعِ غَرَّةٍ / لَا أَسْعُرُ أَتْنِي إِسْتَحَقُّ أَنْ / مُعْرَبًا عَنِ أَمْلِهِ فِي	قسم كلم خاطئ
تَقُولُ أَنَّهُ لِأَعْرَاضِ سِلْمِيَّةٍ تَمَامًا / الْمَوْقِفُ الْآنَ هُوَ إِنَّ مَبْدَأَ الْمُصَالَحَةِ قَائِمٌ / إِذْ أَنَّتْهَا لَنْ / بِأَنَّ يَطْلُبُ عَقْدَ إِجْتِمَاعٍ عَاجِلٍ / أَنْ يَتَّعَمَدَ الْفَقِيدَةَ / مَعَ كُلِّ مَنْ يَهْمُهُ الْأَمْرُ / أَنْ مِنْ يَفْكَرُ فِي نَجَاحِ	تشكيل الأدوات: إن، أن، من
الصَّرَاغُ الْعَرَبِيُّ الْإِسْرَائِيلِيُّ / عَيْدَرِيَّةُ / مَا يُعَادِلُ غَطَاءَ 60.7 يَوْمٍ / مِنْ 61.4 يَوْمٍ فِي يُولْيُو / نَهَائِيَّةِ أَعْسُطُسَ (أب) / حَوْلَ بُنُودِ إِشْكَالِيَّةٍ / أَوْقَفَ عَمِيلَةَ الْبِنَاءِ / إِضَافَةً إِلَى إِعْلَامِ فِلَسْطِينِ وَلِبْنَانَ	أخطاء إملائية لم يتم تصويبها
هُوَ وَأَهْمٌ وَغَيْرُ وَاقِعِي / مِنْ جِهَةٍ أُخْرَى هُنَا خَادِمِ الْحَرَمَيْنِ / غَضُو الْمَجْلِسِ الْوَطْنِيِّ الْفِلَسْطِينِيِّ عَلَى فَيْصَلٍ / وَصَفْتُهُ بِ " إلهام جداً "	تخطئة الكلمات الصحيحة
إِنِّي مُتَفَاجِئٌ / مُتَرَفِّقَةٌ مَعَ خِطَابِ سِيَّاسِي / فِي عَصَبِيَّاتٍ غَرَانِزِيَّةِ	كلمات خارج المعجم
لِشُؤُونِ الْحُجَّاجِ / تَحْدِيدِ سِنَّ الْحُجَّاجِ	نقص تشكيل

On the other hand, error analysis for ADS shows that, on the average, 49% of stem diacritization errors are due to selecting wrong POS, 18% are due to undetected spelling errors, 16% are related to missing diacritics, and 12% are due to diacritizing some particles and function words incorrectly (namely, >n أن, <n إن, and mn من). The rest of errors (~5%) are mainly related to spelling mistakes (there is no out of vocabulary (OOV) words). Figure 12 shows these errors in details and table II lists some examples for each type of errors.

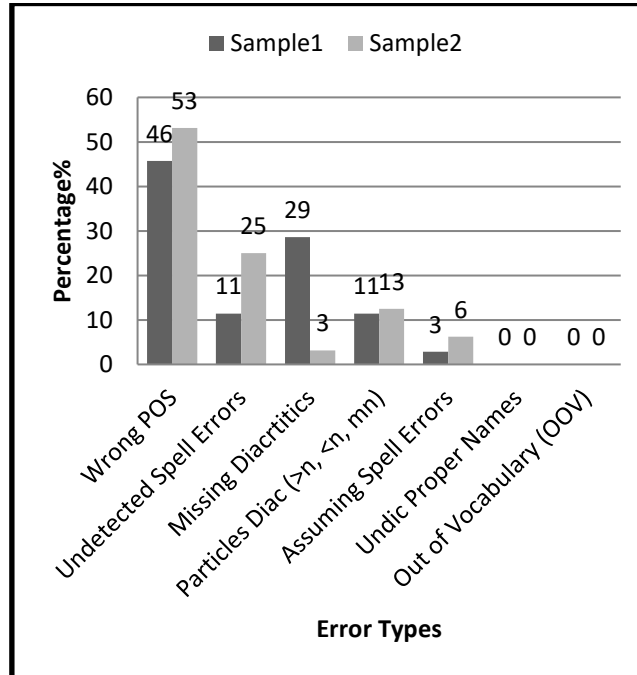


Figure 12: Error Analysis of Stem Diac. for ADS

TABLE II
ANALYSIS of STEM DIAC. ERRORS for ADS

ADS	
أمثلة	الخطأ
التَّقْرِيرَ لَا بُدَّ أَنْ يُنَاقَشَ فِي مَجْلِسٍ / قَرَارِ السُّلْطَةِ سَحَبَ تَقْرِيرٍ / كُلٌّ مَنْ تُثْبِتُ إِدَانَتَهُ	قسم كلم خاطئ
ما يُعادِلُ غِطاءَ 60.7 يَوْمٍ / من 61.4 يَوْمٍ فِي يُولْيُو / أَوْقَفَتِ عَمِيلَةَ البِنَاءِ / إِضَافَةً إِلَى إِعْلَامِ فِلَسْطِينِ وَلُبْنَانَ	أخطاء إملائية لم يتم تصويبها
فِي تَصْرِيحَاتِ ل " الشَّرْقِ الأَوْسَطِ / بِلالِ فَرَحَاتِ ل " الشَّرْقِ الأَوْسَطِ " / مُقَارَنَةً بِ 61 مَلْيُونٍ / يَقْبَلُ الجَائِزَةَ ك " نِداءِ لِلْعَمَلِ/ إف 15 " و 14 طَبَّارًا	نقص تشكيل
المَوْقِفِ الآنَ هُوَ إِنْ مَبْدَأُ المُصَالِحَةِ / قَالَتْ حَرَكَةُ المُقاوَمَةِ الإِسْلامِيَّةِ (حَماس) أَنَّهُ ما	تشكيل الأدوات: إن، أن، من
وَتَقْدِيرِهِ لِلْمَلِكِ عَبْدِ اللَّهِ عَلِيِّ النَّبِّةِ	تخطئة الكلمات الصحيحة
لا يوجد	أعلام غير مشكّلة
لا يوجد	كلمات خارج المعجم

B. Analyzing Case Ending Diac. Errors

Error analysis for MADA shows that, on the average, 28% of case ending diacritization errors are due to incorrectly recognizing subject and object, 15% are due to adjective relation, 14% are due to noun-noun relation "IDafa", 10% are due to conjunction relation, 7% of errors are due to prepositions attached to (or before) nouns, and 5% are due to subject and predicate recognition. The rest of errors (~21%) are mainly related to Inna and Kana sisters, adverbs, "tamyeez", etc. Figure 13 shows these errors in details, and table III lists some examples for each type of errors.

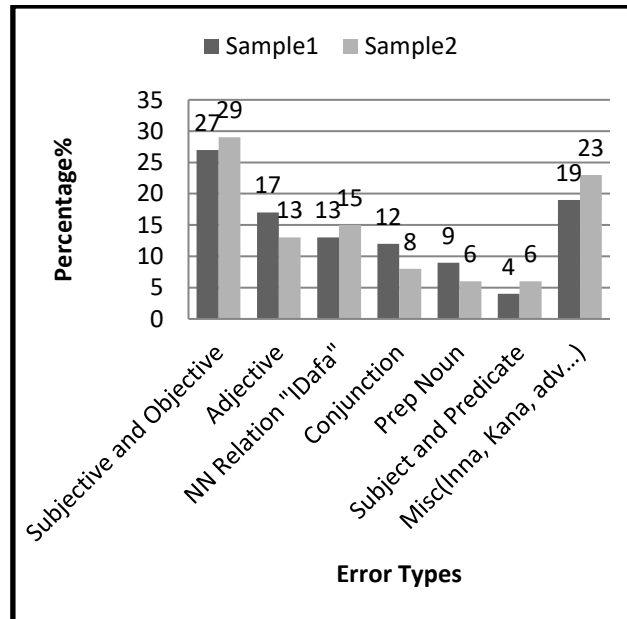


Figure 13: Error Analysis of Case Ending Diac. for MADA

TABLE III
ANALYSIS of CASE ENDING DIAC. ERRORS for MADA

MADA	
الخطأ	أمثلة
الفاعل والمفعول	يُنْتِجُ نِظَائِرُ لِإِعْلَاجٍ / قَدَّمَتَهَا دَوْلٌ مُنْقَرِدَةٌ / خَلَقَتْهَا هَذِهِ الْمُبَادِرَةُ الْإِجْرَامِيَّةُ / وَتَابَعَ الْقَوْلُ / طَالَبَ الْمَكْتَبِ السِّيَاسِيَّ / أَوْضَحَ الْمَكْتَبُ السِّيَاسِيَّ / يَتَوَجَّبُ مُحَاسَبَةً كُلُّ مَنْ تَنَبَّأَ / يَتَحَكَّمُ إِقْبَاعِ الْقِمَّةِ الَّتِي / جَمَعَتِ الْعَاهِلُ السُّعُودِيُّ
الصفة	مَجْلِسُ الْأَمْنِ الدَّوْلِيِّ / وَأَبْلَغَ الْمُجْتَمِعُونَ الرَّئِيسَ الْأَرْمِينِيَّ / التَّلْزَامَ كَامِلًا /
الإضافة	لِتَعْزِيزِ مَحْرُورَاتٍ وَقَوْدٍ / بِمَوْجَةِ إِخْتِجَاجَاتٍ / وَتَوْصِيَّاتٍ تَقْرِيرٍ / وَكَلَّ جِهَاتِ الْإِخْتِصَاصِ / لِعِلَاجِ مَرَضِ السَّرَطَانِ .
العطف	بِمَوْجَةِ إِخْتِجَاجَاتٍ وَاسِعَةٍ وَاعْتِصَامَاتٍ قِبَالَةَ
الجار والمجرور	إِسْتَنْقَظَ عَلَى مُفَاجَأَةٍ / كِتَابَةً عَلَى الْقِيَادَةِ الْأَمِيرِكِيَّةِ / التَّوَاضُعِ فِي تَصْرِيحِهِ / كِنْدَاءٍ لِلْعَمَلِ / تَرْيِدٍ مِنْ الْعِبَاءِ / وَلَيْسَ إِلَى أَقْوَالٍ " / مِنْ وَبَاءٍ إِفْلُونِزَا
المبتدأ والخبر	التَّرْجَمَةَ فِي لُبْنَانَ لَهَا حِسَابَاتٍ / لَهَا صَدَى فِي / هَذِهِ جَائِزَةٌ لِلْمُسْتَقْبَلِ /
إن، كان، الظرف..	وَأَنَّهُ شَخْصِيًّا مُسْتَمِرٌّ فِي النِّضَالِ / أَنَّهُ مِثْلُ بَاقِي الشَّعْبِ / كَانَتْ هُنَاكَ مُقْتَرِحَاتٍ / يُمَكِّنُ أَنْ تَكُونَ أَحَدَ الْبَائِعِينَ / حَتَّى لَا تَكُونَ هَذِهِ الْمُصَالِحَةَ شَخْصِيَّةً

On the other hand, error analysis for MADA shows that, on the average, 36% of case ending diacritization errors are due to incorrectly recognizing subject and object, 17% are due to adjective relation, 13% are due to conjunction relation, 10% are due to subject and predicate recognition, 7% are due to noun-noun relation "IDafa", and 3% are due to prepositions attached to (or before) nouns. The rest of errors (~14%) are mainly related to Inna and Kana sisters, adverbs, "tamyeez", etc. Figure 14 shows these errors in details and table IV lists some examples for each type of errors.

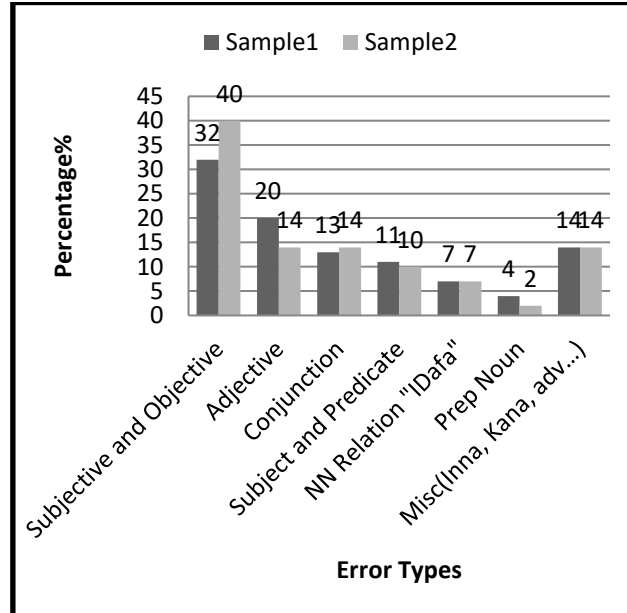


Figure 14: Error Analysis of Case Ending Diac. for ADS

TABLE IV
ANALYSIS of CASE ENDING DIAC. ERRORS for ADS

ADS	
أمثلة	الخطأ
وَتَابِعَ الْقَوْلَ / طَالَبَ الْمَكْتَبَ السِّيَاسِيَّ لِلجَنِبَةِ الشَّعْبِيَّةِ / يَتَوَجَّبُ مُحَاسِبَةً كُلَّ مَنْ تَنَبَّأَ / وَدَعَا الْمَكْتَبَ إِلَى ضَرُورَةِ / سَيَبْتَطِبُ إِنتَاجَ يَوْمِئِثِمِ / تَتَرَقَّبُ إِفْرَاجَاتِ تُؤَدِّي إِلَى /	الفاعل والمفعول
مِنَ الْوَلَايَاتِ الْمُنَحَّدَةِ عَدُوَهَا الْقَدِيمِ / التَّرْكِيبِ - الْأُرْمَنِ الْمُرْمَعِ عَفْدَهُ / وَمُمْتَلِي الطَّوَائِفِ الْأُرْمِينِيَّةِ الْثَّلَاثِ /	الصفة
بِمَوْجَةِ إِحْتِجَاجَاتٍ وَاسِعَةٍ وَاعْتِصَامَاتٍ / وَعَدَدٍ مِنْ الْفَعَالِيَّاتِ الْأُرْمِينِيَّةِ	العطف
الْمَوْقِفِ الْآنَ هُوَ / هُوَ وَاهِمٌ / سَوَاءٌ لَدَى خُلَفَائِهِ / لُبْنَانِ جُزْءٍ مِنْ الْحَالَةِ الْإِقْلِيمِيَّةِ / لَهَا صَدَى فِي / هَذِهِ جَائِزَةِ الْمُسْتَقْبَلِ	المبتدأ والخبر
مُحَاسِبَةً كُلِّ مَنْ / فَوْقَ رَأْسِ أَحَدٍ / مِنْ قَبْلِ بَعْضِ الْجِهَاتِ	الإضافة
ك " نِدَاءٍ لِلْعَمَلِ " / بِوَتِيرَةٍ أَسْرَعَ مِنْ التَّقْدِيرَاتِ / فِي مَسْعَى لِتَعْزِيزِ الرِّقَابَةِ / يَبْسُتَرُ عَلَى أَيِّ فَاسِدٍ	الجار والمجرور
أَنْ يَنْتَهِيَ الْحَقِيقِيَّةِ هِيَ بِنَاءِ فَنْدَلَةٍ نَوَوِيَّةِ / يُمَكِّنُ أَنْ تَكُونَ أَحَدَ الْبَانِعِينَ / سَيَكُونُ لَهُ نَتَاجِجٌ / فِي زِيَارَةِ رَسْمِيَّةٍ لِيَسْتَقْبَلَهُ / دَعْوِي أَكُونُ وَاضِحًا / تَوَقَّعَاتِ اِقْتِصَادِيَّةِ أَكْثَرَ تَقَاوُلًا / 252.6 مِلْيُونِ رِيَالٍ	إن، كان، الظرف..

C. Calculating WER and DER

For the same samples, we calculated manually WER and DER for MADA and ADS. We found that MADA achieved an average WER of 16.93% and an average DER of 3.4% compared to ADS which achieved a WER of 2.57% and a DER of 0.4%. This is shown in Figure 15.

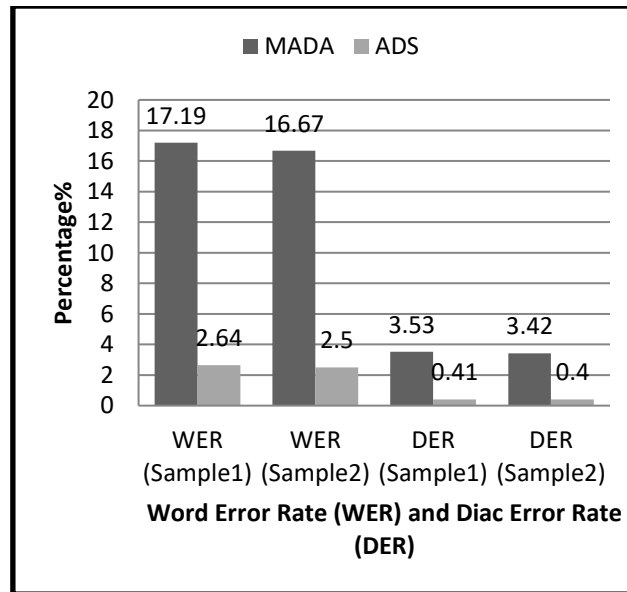


Figure 15: WER and DER for MADA and ADS

It is observed that MADA has common problems that can be easily enhanced to minimize both WER and DER. These problems can be classified as a missing diacritic in the following cases:

- “moon Lam القمرية اللام (ex: الإيراني Al<irAniy~)
- letters before vowels (ex: مَحْمُودِ maHomwd).
- last letter in function words with/out suffixes (ex: مِنْ min, عَنْهُ Eanhu)
- last letter of some suffixes(ex: حُقُوقِهِم Huqukihim)
- “feminine Taa تاء التانيث المفتوحة (ex: عَرَضَتْ EarDat)

The following figure shows these missing and wrong diacritics for MADA and ADS for an arbitrary sentence.

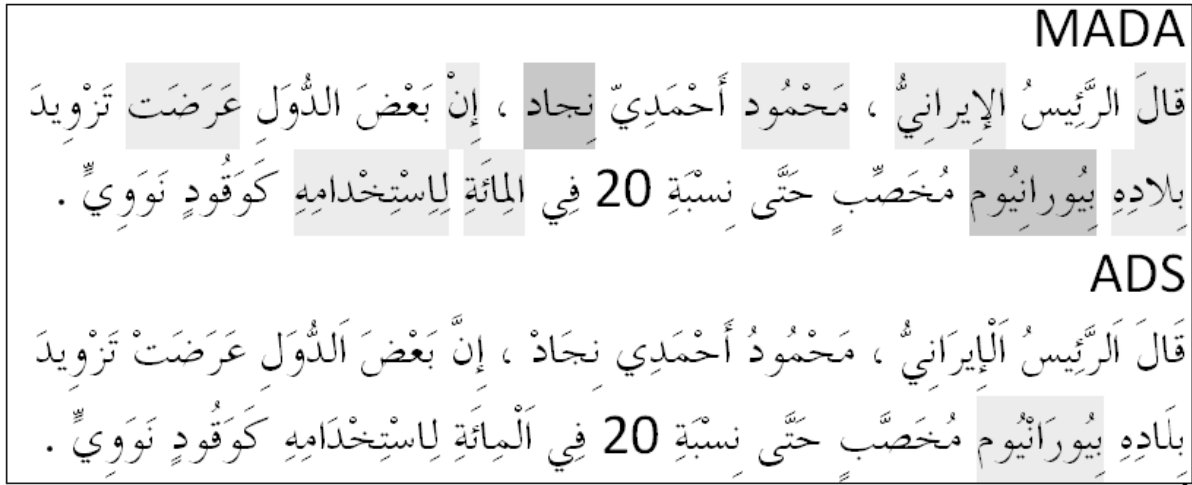


Figure 16: Highlighting Diacritization Errors

Because there is no standard test bench for measuring WER and DER, we just summarize in table V some reported evaluation experiments for different diacritizers.

TABLE V
WER% and DER% (IN ORDER) for SOME DIACRITIZERS

Evaluator	MADA	Zitouni	Sakhr ADS	RDI	Shaalán	KACST
MADA (Habash, N.)	14.9 4.8					
Zitouni (Zitouni, I.)		18.0 5.5				
Sakhr ADS (Mubarak, H.)	16.9 3.4		2.6 0.4			
RDI (Rashwan, M.)	14.9 5.5	18.0 7.9		12.5 3.1		
Shaalán (Shaalán, K.)					11.8 3.2	
KACST (Alghamdi, M.)						26.0 9.2

6 CONCLUSIONS

In this paper, we presented Sakhr Arabic disambiguation system (ADS) which resolves morphological, lexical, and semantic ambiguity in Arabic texts. We compared the ADS diacritization with the best diacritization system that is reported in the literature so far (MADA). We analyzed errors in diacritizing stem and case ending for both engines, and measured word error rate (WER) and diacritic error rate (DER). We recommend here to have a standard test bench for evaluating different Arabic diacritizers, and also to measure both stem errors and case ending errors separately as their impacts on word meaning are not the same.

REFERENCES

- [1] M. Alghamdi and Z. Muzaffar, *KACST Arabic Diacritizer*. The First International Symposium on Computers and Arabic Language, 2007.
- [2] M. Diab, M. Ghoneim, and N. Habash, *Arabic Diacritization in the Context of Statistical Machine Translation*, MT Summit XI, Copenhagen, Denmark, 2007.
- [3] M. Elshafei, H. Almuhtasib, and M. Alghamdi, *Machine Generation of Arabic Diacritical Marks*. The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing. Las Vegas, USA, 2006.
- [4] A. Farghaly, and K. Shaalan, *Arabic Natural Language Processing: Challenges and Solutions*. ACM Transactions on Asian Language Information, 2009
- [5] N. Habash, and O. Rambow, *Arabic Diacritization Through Full Morphological Tagging*, The North American Chapter of the Association for Computational Linguistics (NAACL).Rochester, New York, 2007.
- [6] M. Maamouri, A. Bies, and S. Kulick, *Diacritization: A Challenge to Arabic Treebank Annotation and Parsing*. In Proceedings of the Conference of the Machine Translation SIG of the British Computer Society, 2006.
- [7] H. Mubarak, K. Shaban, and F. Adel, *Lexical and Morphological Statistics of an Arabic POS-Tagged Corpus*. The 9th Conference on Language Engineering, Cairo, Egypt, 2009.
- [8] M. Rashwan, M. Al-Badrashiny, M. Attia, S. Abdou, and A. Rafea, *A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features*, IEEE Transactions on Audio, Speech, and Language Processing, 2011.
- [9] K. Shaalan, H. Abo Bakr, I. Ziedan, *A Hybrid Approach for Building Arabic Diacritizer*, Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, Association for Computational Linguistics. Athens, Greece, 2009.
- [10] I. Zitouni, J. S. Sorensen, and R. Sarikaya, *Maximum Entropy Based Restoration of Arabic Diacritics*, in Proceedings of ACL'06, 2006.

Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas

Sameh AlAnsary

Department of Phonetics and Linguistics, Faculty of Arts, Alexandria University

ElShatby, Alexandria, Egypt.

Bibliotheca Alexandrina, ElShatby, Alexandria, Egypt.

Sameh.alansary@bibalex.org

Abstract—Interlingua-based machine translation is probably the most attractive among the three classic approaches to MT. Early pioneers as well as current researchers experimented with this approach and produced some very stimulating methodologies to reaching such a language-independent framework. In this paper, we shall briefly review some of the most renowned endeavours in interlingua-based machine translation and bring into view how the latest of which; the Universal Networking Language (UNL) differs and compares to these other systems.

1 INTRODUCTION

Generally, three classic approaches have been acknowledged in the field of Machine Translation; Direct, Transfer and Interlingua. The Direct approach is mainly a lexicon-based approach in which a computer program performs a word-for-word substitution (with some local adjustment) between language pairs using a large bilingual dictionary “Ref. [1]”.

The Transfer approach operates over three stages: analysis, transfer and generation. First, the SL text is parsed into an source-language-specific intermediate syntactic structure. Then, linguistic rules specific to the language pair transform this representation into an equivalent representation in the target language. Finally, the final target language text is generated “Ref. [2]”.

The Interlingua approach, on the other hand, is based on “the argument that MT must go beyond purely linguistic information (syntax and semantics) and involve an ‘understanding’ of the content of texts” “Ref. [1]”. Interlingua-based translation is divided into two monolingual components: analyzing the SL text into an abstract universal language-independent representation of meaning (the interlingua), and generating this meaning using the lexical units and the syntactic constructions of the target language.

2 INTERLINGUA: DEFINITIONS AND CHARACTERISTICS

The motivation behind devising an interlingua was the long-lived belief that while languages differ greatly in their “surface structures”, they all share a common “deep structure”. Hence arose the idea of creating a universal representation capable of conveying this deep structure while enjoying the regularity and predictability natural languages lack.

In order to be capable of representing natural language content, an interlingua should be, first, unambiguous; it should be more explicit even than the natural language it is representing “Ref [1]”. Second, it should represent the full content of the input text; its morphological, syntactic, semantic and even pragmatic characteristics “Ref. [3]”. Third, it should be universal, capable of representing the abstract meaning of any text, belonging to any domain or language. Fourth, an interlingua should represent the content of the input alone and not be influenced by the formal representation of the content in the SL text “Ref. [4]”. Fifth and finally, the interlingua should be independent of both the SL and the TL; analysis should be SL-specific and not oriented to any particular TL, and likewise should be the generation “Ref. [5]”.

The advantages of using such an approach include economy, modularity, localization, back-translation possibility and potential uses in other NLP-related areas such as cross-lingual information retrieval, summarization, rephrasing and question answering “Ref. [3], [1], [4], [6]”.

3 SOME WELL-KNOWN INTERLINGUA-BASED SYSTEMS

Despite its numerous advantages, the interlingua approach is probably the least used among the three classic approaches. However, many research projects have produced quite promising prototypes. The following section briefly reviews three of the most renowned interlingua-based machine translation projects.

A. DLT

DLT stands for Distributed Language Translation, a research project developed in Utrecht, The Netherlands. Preliminary research in the project began as early as 1979. In 1984, DLT entered a six-year project to build an MT system capable of translating from simplified English into French. However, in 1990, the DLT pilot project came to an end “Ref. [7]” after receiving a fair amount of publicity.

DLT is an interactive system developed to operate over computer networks. Translation is distributed between two independent terminals; one for the analysis and another for generation. In the DLT system, the intermediate representation (the interlingua) is a ready-made logical language with supposedly standardized rules for vocabulary and structures; i.e. Esperanto.

Semantic and Pragmatic knowledge constitute the language-independent component of the system and is completely handled in the intermediate stages of forming the Esperanto representation. Language-specific information, on the other hand, is purely syntactic and is developed for a specific pair of languages, in one translation direction only; from English to Esperanto, for instance “Ref. [1]”.

The text entered at one terminal is syntactically parsed into dependency trees. In case of syntactic ambiguity, the parser produces all possible alternative trees regardless of their semantic probability “Ref. [8]”. “The result is a (sometimes large) number of 'formally possible' parallel translations” “Ref. [9]”. Then, rules replace SL words with their Esperanto equivalents (all possible alternatives), and English syntactic labels with Esperanto ones.

So far, all candidate parses are equally probable. To choose one, first, the system consults the Lexical Knowledge Bank (LKB) which is a database containing pairs of content words linked by a connector (see figure 1) “Ref. [1]”. Its role is to indicate which word is most likely to appear in the given context.

<i>ĉambro a hela</i>	'light room'
<i>ĉambro a komforta</i>	'comfortable room'
<i>ĉambro a komuna</i>	'common room'
<i>ĉambro a nuda</i>	'bare room'
etc.	

Figure 1: A sample from DLT's Lexical Knowledge Base (LKB)

If no exact match was found in the LKB, an algorithm called SWESIL ranks the possible alternatives according to their semantic proximity. If, after all, the system was not able to conclusively choose one by itself, a machine-initiated disambiguation dialogue presents the operator, in his native language, with the phrases or sentences requiring disambiguation, on which he/she may choose one of the possible interpretations listed on the screen “Ref. [9]”. Finally, the chosen tree is regularized and linearized into a plain Esperanto text as shown in figure 2 “Ref. [1]” which is subsequently sent to the Decoding terminal.

Al multnaciaj entreprenoj asignajis subvencioj.

Figure 2: The Esperanto representation of the English sentence "Multinationals were allocated grants"

The Decoding terminal starts by parsing the Esperanto text into a dependency tree, and replacing Esperanto lexical items by those of the target language. However, because there are usually several words that are possible translations to a single Esperanto lexical item, the Metataxor generates several target dependency trees from a single Esperanto tree.

Disambiguation, in this half of the process, requires bilingual information; an Esperanto to target language bilingual dictionary that contains Esperanto word pairs as contextual clues for the plausibility of a word in a given context (see figure 3) “Ref. [10]”. In addition, there can be no interaction with the receiving user.

Esperanto entry word	Semantic relator (an Esperanto morpheme)	Disambiguating contexts: Esperanto words (with morpheme tokens), here illustrated with approximative English glosses	French word with syntactic word class
<i>akr'a</i>	<i>a</i>	<i>dolor'o, mal'varm'o, riproc'o'j, vor'o'j, romp'o, eĝ'o</i> 'pain', 'cold', 'blames', 'words', 'break', 'edge'	<i>vi</i> /ADJ
<i>akr'a</i>	<i>a</i>	<i>naz'o, orel'o'j, tur'o</i> 'nose', 'ears', 'tower'	<i>pointu</i> /ADJ
<i>akr'a</i>	<i>a</i>	<i>spic'o, pipr'o, brand'o</i> 'spice', 'pepper', 'brandy'	<i>for</i> /ADJ

Figure 3: A sample from the Esperanto to French bilingual dictionary

If no exact match was to be found in the bilingual dictionary, proximity scores are again calculated using SWESIL. The target dependency tree is finally linearized and adjusted to form a readable text to be received by the target user. The overall design of the DLT system is shown in figure 4 “Ref. [1]”.

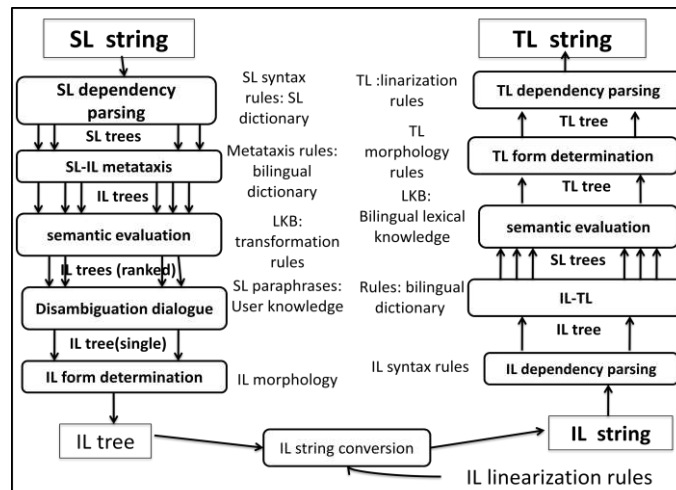


Figure 4: The overall design of the DLT system

B. UNITRAN

The name UNITRAN stands for UNiversal TRANslator; a translation system developed at Massachusetts Institute of Technology. The system operates bidirectionally between Spanish and English. However, other languages may be added by setting the parameters that fit them “Ref. [11]”.

The UNITRAN system comprises two main components between which processing tasks are divided; the syntactic component and the lexical-semantic component. The syntactic component is based on the Government and Binding theory, it is responsible for handling the language-specific syntactic differences by accepting and producing grammatically correct sentences. The syntactic component is composed of a set of parameters associated with universal principles. These parameters are built-in in the analyzer and generator to be set according to the values of the language being processed. Thus, the analyzer and generator used are the same for all languages. The lexical-semantic component, on the other hand, is based on the Lexical Conceptual Structure theory, it contains the information necessary to provide a conceptual form (the LCS) to underlie the source language sentence, and to match it to the appropriate target-language lexical items “Ref. [11], [12]”.

The intermediate representation (the LCS) relies on a set of primitives that serve as the basic units of meaning such as event, state, property...etc. “Ref. [12]”.

First, the operator sets the analyzer parameters to suit the values of the source language. For example, the “null subject” parameter has to be set to “yes” for Spanish and Italian...etc., but to “no” for English and German...etc. This is done through a menu operation.

The processing, then, begins by the syntactic component parsing a morphologically analyzed input into a tree showing the structural relations between constituents. Then, the lexical-semantic component maps each source word onto its corresponding LCS. The resulting LCS forms are subsequently merged into a single LCS (the composed LCS) which is the interlingua representation underlying the whole input sentence (see figure 5) “Ref. [12]”.

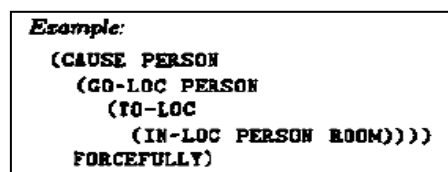


Figure 5: The composed LCS underlying the sentence "John broke into the room"

The second stage is substitution. Each node in the composed LCS is mapped onto a target language word and the resulting LCS is mapped onto the syntactic realization of the target language sentence.

After setting the generator’s parameters to meet the requirements of the target language, the generation process start by performing structural movement and generating the correct morphological forms of the target sentence’s constituents. Figure 6 shows the overall design of the UNITRAN translation system “Ref. [11], [12]”.

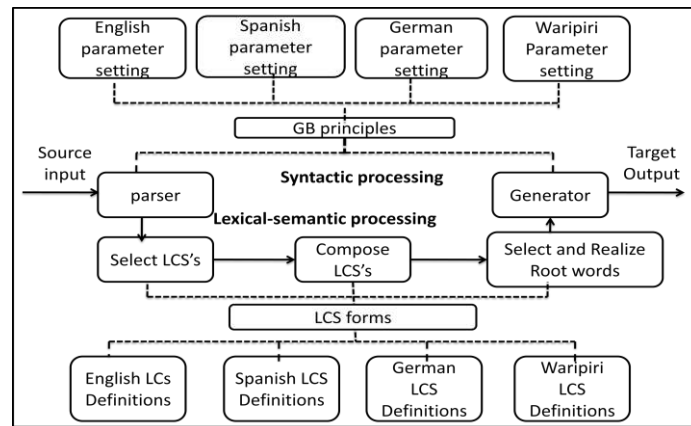


Figure 6: the overall design of UNITRAN

C. KANT

The KANT (Knowledge-based, Accurate Natural-Language Translation) system has been developed at Carnegie-Melon University (CMU) in Pennsylvania, USA in 1989 “Ref. [13]”. KANT is the only interlingua-based MT system to be operational commercially. It has been used in translating English technical documents into French, Spanish and German. The addition of more target languages such as Portuguese, Italian, Russian, Chinese and Turkish is under research “Ref. [4]”. The KANT prototype has also been used in generating Japanese and German “Ref. [14]”.

KANT is a sublanguage translation system; it is used by large manufacturers to translate their technical documentation from English into several target languages “Ref. [13]”. “Though the analysis component must support generation in multiple languages, it currently handles only one source language, and therefore can tolerate a slight degree of source language dependence” “Ref. [15]”.

The system codes for analysis and generation are language-independent whereas the specific knowledge required to process a certain language (grammars and lexicons) is developed separately for each language “Ref. [13]”.

The first stage in the translation process is concerned with authoring the input. KANT is designed to translate only a well-defined subset of source language “constrained both by the domain from which the source texts are drawn (e.g. service information for heavy machinery), and by general restrictions” that are put on the vocabulary and structures of input language “Ref. [15]”. Kant’s vocabulary (non-domain specific) is limited to a basic vocabulary of about 14,000 distinct word senses while domain-specific technical terms are limited to a pre-defined vocabulary “Ref. [16]”, approximately 60,000 words and phrases for heavy equipment manuals “Ref. [17]”. Structural restrictions, on the other hand, attempt to limit the use of constructions that would create difficulties in parsing such as the use of relative clauses with an explicit relative pronoun rather than reduced relative clauses “Ref. [18]”.

In the first processing stage of knowledge-based parsing, the source text is processed using the source language grammar and lexicon to produce a Source F-Structure (a grammatical functional structure) for each sentence. Kant uses an explicit and very restricted domain model-based semantic restrictions to resolve ambiguity (e.g. phrase attachments). An example of these semantic restrictions is shown in figure 7 “Ref. [14]”.

```
(*B-CLEAN
(is-a *EVENT)
(agent *USER)
(theme *PHYSICAL-LOCATION
*PHYSICAL-OBJECT)
(instrument *O-CLEANING-INSTRUMENT))
```

Figure 7: Kant’s semantic restrictions on the English verb “clean”

In the Interpretation stage, mapping rules map lexical items onto semantic concepts, and syntactic arguments onto semantic roles, forming the intermediate representation (see figure 8) “Ref. [15]”. The interlingua representation comprises information from all necessary levels of linguistic analysis; lexical, syntactic, semantic and pragmatic

```

"The primary power supply component will supply the necessary 240 Volts DC to the input lead." =>

(*a-supply
 (tense future)
 (mood declarative)
 (punctuation period)
 (source (*o-power-supply-component
 (reference definite)
 (number singular)
 (attribute (*p-primary))))
 (theme (*u-volt-dc
 (reference definite)
 (number plural)
 (attribute (*p-necessary))
 (quantity
 (*c-decimal-number
 (integer "240")
 (number-type cardinal)
 (number-form numeric))))))
 (goal_to (*o-input-lead
 (reference definite)
 (number singular))))

```

Figure 8: The interlingua representation for a sentence from a television repair manual

In the generation stage, target mapping rules indicate how the interlingua representation maps onto the appropriate Target F-Structure. The overall architecture of the Kant translation system is shown in figure 9 “Ref. [17]”.

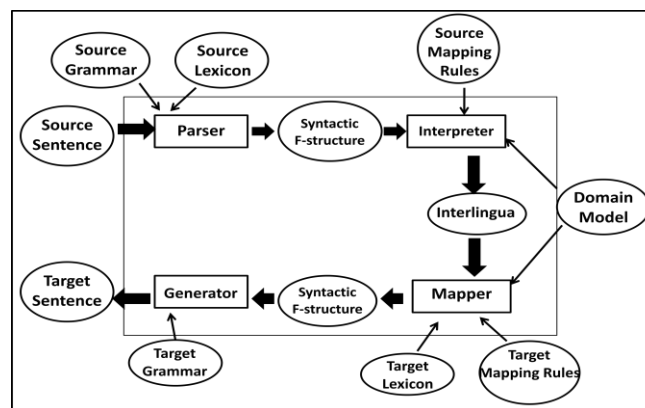


Figure 9: The run-time architecture of KANT

D. UNL

The Universal Networking Language project was launched in 1996 at the Institute of Advanced Studies of the United Nations University (UNU/IAS), Tokyo, Japan. In January 2001, the United Nations University set up an autonomous non-profit organization in Geneva, Switzerland to be responsible for the development and management of UNL; the Universal Networking Digital Language (UNDL) Foundation. In addition, 17 language centers all over the world are working on the development of the UNL resources necessary for incorporating their native language into the UNL program. Among these centers are the Arabic UNL center in Alexandria, Egypt (<http://www.bibalex.org/unl>), the Spanish center in Madrid, Spain (www.vai.dia.fi.upm.es) and the Russian center in Saint Petersburg, Russia (www.unl.ru).

The mission of the UNL program is to overcome the language barrier and enable all peoples to generate, and have access to, information and knowledge in their native languages and cultures by coding, storing and disseminating human knowledge, in any given domain, in a language-independent format that represents only the core content and abstracts away from the particular characteristics of the original language in which it was expressed¹.

UNL is not intended to be an auxiliary language such as Esperanto, Interlingua, Ido or others, it is rather a formal artificial language that replicates the functions of natural language in communication, but is, nevertheless, designed for computers rather than humans. People should use UNL in “communication” in the same subtle manner they do with other procedural languages such as HTML.

The UNL program has passed through several stages of development, the third and latest of which is the UNL+3 project; a three-year project to advance the long-term mission of the UNDL and make the UNL fully operational by the end of 20112. In this phase, the linguistic infrastructure has been developed using the x-bar theory. Accordingly, the analysis and generation

¹ More information about the UNDL, its ideology and its goals is available at <http://www.unl.org>

² More information about UNL+3 is available at www.unlweb.net.

processes pass over five stages, rather than the direct approach adopted in the previous approaches, to help yield more accurate results.

The main bulk of the UNL system is language-independent. The engines' codes necessary for converting natural language input into UNL (UNLization) and converting UNL into natural target language (NLization) are the same whatever the input or output language may be. In addition, information on the semantic abstract concepts (Universal Words or UWs) depicted by different cultures are organized hierarchically in a common ontology called the UNL Knowledge Base (UNLKB). These UWs are only expressed in English for the sake of readability. Figure 10 shows samples from the UNLKB.



Figure 10: Samples from the UNLKB

Language-dependent resources, on the other hand, are developed by the language center of the respective language. They include the lexicon that maps natural language lexical items onto universal concepts (Universal Words or UWs) and vice versa, and the grammar rules that determine well-formedness standards.

Translation takes place over two completely independent processes; UNLization and NLization. The language-independent UNLization tool (IAN) converts natural language input into UNL format through five phases. First, the natural language list is processed to identify the abstract concepts represented by the words in the input sentence using the language's word dictionary. Second, these constituents are parsed into a surface syntactic tree. Third, the surface tree is analyzed on a deeper level to form the deep syntactic tree. Fourth, the syntactic tree is transformed into a semantic network and finally the network is post-edited for any modifications that would make the resulting semantic network more accurate. Figure 11 shows an example of an UNLization rule.

(ART,def,%x)(N,%y):=(N,%y,@def);

Figure 11: The UNLization rule that substitutes the definite article "ال" into a "@def" attribute

The result is a semantic network (called the UNL expression). A UNL expression is the input for the NLization process. UNL expressions represent all aspects of input content; semantic, syntactic, pragmatic, format...etc. Semantic information about the abstract concepts themselves is stored with each UW. As for the semantic links that tie these concepts in a given sentence, they are expressed via Relations. Relations are three letter symbols expressing an ontological relation such as "icl" (a kind of), a thematic relation such as "agt" (agent), or a logical relation such as "and" "Ref. [19]". Note that these Relations are entirely semantic and are not influenced by the syntactic roles of the constituents in the sentence being processed.

On the other hand, grammatical information such as Tense, Aspect, Person, and Number...etc. is encoded in the form of Attributes of linguistic features. Attributes are tags that annotate a particular word in the UNL expression such as "@past", "@progressive"...etc. Attributes also express contextual and subjective information such as "@discontented", and "@insistence". Some information about the formatting of the original co-text is also encoded in Attributes such as "@parenthesis" and "@title"³ "Ref. [19]". Figure 12 shows an example of a UNL graph. Linguistic features on the other hand are extracted from the UNL tagset. The UNL tagset is a standardized repository containing tags for some specific and pervasive grammatical phenomena. Many of those linguistic constants have been proposed to the Data Category Registry (ISO 12620), and represent widely accepted linguistic concepts. This tagset helps standardize and harmonize the UNL resources so as to make them understandable and exchangeable as possible. Tags in the tagset are used to mark each entry in a language lexicon with all the linguistic information it carries; such as number, gender, semantic typology, register, etc. The final UNL network will look as shown in figure 12 while the equivalent UNL expression is shown in figure 13.

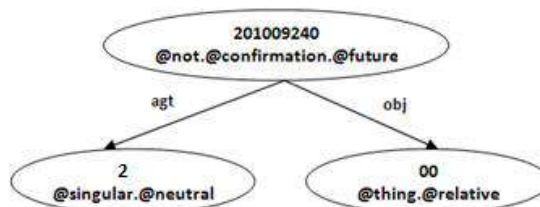


Figure 12: The UNL graph representing the sentence "you won't say that will you?"

³ The complete set of UNL specifications and components is also available at <http://www.undl.org/>

```

agt(201009240:XC.@future.@not.@confirmation.
@entry,      00:DF.@2.@singular)
obj(201009240:XC.@future.@not.@confirmation.
@entry,      00:DM.@thing.@relative)

```

Figure 13: The UNL expression for the UNL network in figure 12.

The Deconversion process begins after receiving the UNL expressions of the text to be translated. The language-independent NLization tool (EUGENE) uses the target language word dictionary to transform it into a directed hyper-graph structure called the Node-net. The NLization process passes through five phases similar to the UNLization process but in the reverse order. First, the UNL network is edited in order to make it more suitable for translation. Second, the network is transformed into a deep syntactic structure from which the surface structure is extracted in the third phase. In the fourth phase the tree structure is linearized into a list structure and finally this list is post-edited for morphological adjustments to produce a well-formed comprehensible natural language sentence. Figure 14 shows an NLization rule.

```

(%x,M504,DUA,NOM):=(%x,-
M504,+FLX(DUA&NOM:="ان"<0,"ت":"ة"))

```

Figure 14: The rule for generating the dual form from a nominative noun ending with a “ة” by replacing the final “ة” With “ت” and adding “ان” at the end

The overall architecture of the UNL translation process is shown in figure 15.

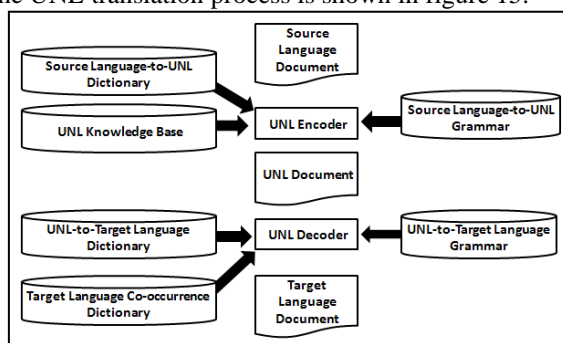


Figure 15: The overall architecture of the UNL system

In the current phase of UNL development; UNL+3, specifications have been modified in order to cover even more linguistic phenomena, and to handle some of the problems in the earlier stages. The new project also offers a free and open virtual learning environment (VALERIE) for those wishing to contribute to the development of such a massive project⁴ in addition to the UNLarium which is an open-source web-based development environment where registered users are able to create, edit, share, search, export and download lexical and grammatical resources that have been provided by other users and in other languages⁵ “Ref. [20], [21]”.

Although Machine Translation is one of the possible and more obvious and promising uses of UNL, it is not the only area in NLP where UNL can prove useful. As it offers a complete understanding of natural language content, UNL can serve areas such as summarization and text simplification. Moreover, by providing a language-neutral representation of meaning, UNL can dramatically improve our ability to search for and find information, thus, helping areas such as information retrieval and others.

The Universal Networking Language has already proved its efficiency in several projects such as encoding the contents of 25 English documents from the Encyclopedia of Life Support Systems (EOLSS) in UNL, and successfully generating their Arabic, French, Japanese, Russian and Spanish counterparts. “Ref. [22], [23]”. The output of this project has been evaluated qualitatively and statistically and the results were significantly higher than those of Google, Babylon and Sakhr’s Tarjim “Ref. [24]”. In addition, UNL has been used in creating a prototype language-independent Library Information System (LIS) that provides the resources necessary for the generation of books’ metadata into at least six languages other than the original Arabic “Ref. [25]”.

⁴ Available at www.unlweb.net/valerie/

⁵ Available at (www.unlweb.net/unlarium/)

4 DISCUSSION

All of the previous projects exhibit intriguing approaches to defining an abstract language-independent format for knowledge representation. However, they vary in the degree of language-independency, the complexity through which they achieve such a representation and in their capabilities. Most of the previous systems incorporate a stage of syntactic parsing which leads to a semantic mapping of the resulting syntactic tree. UNL also uses such stages but instead of two stages, UNL uses five gradual stages to analyze the natural language sentence morphologically, on the surface syntactic level, on the deep syntactic level and semantically, and vice versa in generation. This leads to far greater accuracy in the understanding of natural language.

“the interlingua approach necessarily requires complete resolution of all ambiguities in the SL text so that translation into any other language is possible” “Ref. [14]”. Hence, a large section of processing stages in most interlingua-based MT system is devoted to disambiguating the input to form the interlingua, and in some cases, disambiguating the intermediate representation to derive the output (as in the DLT system). In other cases, the system enforces very strict limitations on input language and imposes precise semantic restrictions on the abstract concepts to avoid prolonged processing (as the case in Kant). As a last resort, some systems turn to interactive communication with the user(s) such as DLT. UNL has largely avoided such intricate procedures by using a quite unambiguous intermediate representation. In UNL, a word can never have more than one conceptual representation; they are clearly distinguished by the ID number that represents their exact contextual meaning. For example, “bank” meaning “a financial institution that accepts deposits and channels the money into lending activities” is clearly differentiated from “bank” meaning “sloping land (especially the slope beside a body of water)” by means of the Universal Words “108420278” and “109213565”, respectively. However, the UNL system does employ disambiguation techniques on the word, syntactic tree and semantic network levels, but these techniques are entirely optional, when not used, the system can still output acceptable results.

Moreover, most interlingua-based MT systems miss one aspect of meaning or another such as UNITRAN that does not incorporate the notion of grammatical aspect “Ref. [12]”. UNL, on the other hand, tries to convey all aspects of meaning in its intermediate representation; semantic, syntactic, pragmatic, subjectivity, format ...etc. Yet, UNL developers acknowledge that the “subtleties of intention and interpretation make the “full meaning” [. . .] too variable and subjective for any systematic treatment”. Hence, it avoids the mistake of “trying to represent the “full meaning” of sentences or texts, targeting instead the “core” or “consensual” meaning that is most often attributed to them”. It is also not committed to replicate the lexical and the syntactic choices of the original input and can be, therefore, regarded as an interpretation rather than a translation⁶.

Mapping natural language onto an unambiguous conceptual representation is indeed quite challenging, which is why several projects attempted to curb the difficulty by either limiting input texts to specific domains (such as Kant) or controlling input language vocabulary and structures (such as Kant and DLT’s prototype). UNL, however, does not put any kind of restriction on input texts or language; nevertheless, “much of the subtlety of poetry, metaphor, figurative language, inuendo and other complex, indirect communicative behaviors is beyond the current scope and goals of the UNL”. Instead, it focuses on “direct communicative behavior” which accounts for “much or most of human communication in practical, day-to-day settings”⁷.

Although an interlingua should, in theory, be universal, no interlingua-based system has ever intermediated between more than 10 languages. Still, UNL’s mission is to eradicate language barriers by intermediating between all natural languages and has already started by incorporating 17 languages. UNL, as a non-profit project, would make possible the instant generation of various target-language versions of such a vital source of knowledge such as the internet, upon request, if WebPages were to contain a UNL representation of its content along with the original language.

UNL is not simply an intermediate representation; it is a full-scale language for machines. This means that its uses go beyond the task of translation as mentioned earlier. Besides, it can represent any imaginable concept because, unlike a system such as UNITRAN which builds concepts from a limited set of primitives “Ref. [12]”, it makes use of dozens of semantic Relations and Attributes to exactly convey the intended meaning. Another system such as the DLT uses a regularized “human” language “with its own lexical items and syntactic rules” which “caused translation in the DLT system to be sometimes viewed as, in fact, two translation processes rather than one” “Ref. [1]”.

Unfortunately, due to its challenging nature, most interlingua-based system never makes it beyond the research phase. UNL, on the other hand, is no more a pilot project; it was launched in 1996 and has been ever since subject to constant developments and enhancements under the auspices of the UNDL foundation and the United Nations. The most recent development (the UNL+3) recruits even more participants by offering a free learning environment and promotes integration by providing an open-source environment for developers to share their resources. Besides, UNL has also been successfully used in numerous projects and its output is constantly subject to evaluation.

⁶ This excerpt is taken from http://www.unlweb.net/wiki/index.php/Introduction_to_UNL

⁷ From http://www.unlweb.net/wiki/index.php/Introduction_to_UNL

5 CONCLUSION

A language-neutral representation of meaning has always been the dream of MT researchers. Although it is one of the oldest approaches in the field, very few systems have ever attained international recognition. This paper describes three of the most referred to systems as pioneers in devising an interlingua-based system, briefly examining their designs and characteristic features and how a more modern fourth system; UNL, has succeeded in mending some of their imperfections that impeded reaching the ultimate goal of bringing down the language barriers.

REFERENCES

- [1] W. J. Hutchins, and H. L. Somers, *An Introduction to Machine Translation*, (chapter 1, 4, 17) (chapter 1 p.8) London Academic Press Limited, 1992.
- [2] S. Nirenburg and Y. Wilks, "Machine Translation", *Advances in Computer*, vol. 52, pp. 160-189, 2000.
- [3] A. Lampert, "Interlingua in Machine Translation", Technical Report, 2004.
- [4] Bonnie J. Dorr, E. Hovy and L. Levin, "Machine Translation: Interlingual Methods", *Encyclopedia of Language and Linguistics*. 2nd ed., Brown, Keith (ed.), 2004.
- [5] W. J. Hutchins, "Machine translation: a brief history", *Concise history of the language sciences: from the Sumerians to the cognitivists*. E.F.K.Koerner and R.E.Asher (eds.), Pergamon, pp.431-445, 1995.
- [6] H. Uchida and M. Zhu, "Interlingua for Multilingual Machine Translation", in *Proceedings of the Machine Translation Summit IV*, Kobe, Japan, July 20-22, 1993.
- [7] T. Witkam, "*History and Heritage of the DLT (Distributed Language Translation) project*". [Utrecht, The Netherlands: private publication, 2006].
- [8] D. Maxwell, K. Schubert and T. Witkam (eds.), *New Directions in Machine Translation*, (chapter 8), Foris Publications, Dordrecht, Holland, 1988.
- [9] T. Witkam, "DLT — An Industrial R&D Project for Multilingual MT", in *Proceedings of Proceedings of the 12th International Conference on Computational Linguistics (COLING 1988)*, Budapest, 1988.
- [10] D. Maxwell, K. Schubert and T. Witkam (eds.), *New Directions in Machine Translation*, (chapter 8), Foris Publications, Dordrecht, Holland, 1988.
- [11] Bonnie J. Dorr, "UNITRAN: An Interlingua Approach to Machine Translation". in *Proceedings of the 6th Conference of the American Association of Artificial Intelligence*, Seattle, Washington, 1987.
- [12] Bonnie J. Dorr, "A cross-linguistic approach to translation". in *Proceedings of 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, Linguistics Research Center, University of Texas, Texas, 11-13 June, 1990.
- [13] E. H. Nyberg, T. Mitamura and J. Carbonell, "The KANT Machine Translation System: From R&D to Initial Deployment¹", in *Proceedings of LISA (The Library and Information Services in Astronomy) Workshop on Integrating Advanced Translation Technology*, Hyatt Regency Crystal City, Washington D.C., June 3-4, 1997.
- [14] T. Mitamura, E. H. Nyberg III and J. G. Carbonell, "An Efficient Interlingua Translation System for Multi-lingual Document Production", in *Proceedings of Machine Translation Summit III*, Washington D.C., The United States, July 2-4, 1991.
- [15] Deryle W. Lonsdale, A. M. Franz and J. R. R. Leavitt. "Large Scale Machine Translation: An Interlingua Approach". in *Proceedings of the 7th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Austin, Texas, The United States. 1994.
- [16] E. H. Nyberg and T. Mitamura, "The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains," in *Proceedings of the International Conference on Computation Linguistics*, (COLING 1992), Nantes, France, July, 1992.
- [17] T. Mitamura, E. H. Nyberg 3rd and J. G. Carbonell, "Automated Corpus Analysis and the Acquisition of Large, Multi-Lingual Knowledge Bases for MT¹", in *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation*, Kyoto, Japan, July 14-16, 1993.
- [18] T. Mitamura, "Controlled Language for Multilingual Machine Translation¹", in *Proceedings of Machine Translation Summit VII*, Singapore, September 13-17, 1999.
- [19] H. Uchida and M. Zhu, "UNL2005 for Providing Knowledge Infrastructure", in *Proceedings of the Semantic Computing Workshop (SeC2005)*, Chiba, Japan, 2005
- [20] S. Alansary, M. Nagi and N. Adly, "UNL+3: The Gateway to a Fully Operational UNL System", in *Proceedings of the 10th Egyptian Society of Language Engineering Conference (ESOLEC 2010)*, Ain Shams University, Cairo, Egypt, December 15 – 16, 2010.

- [21] S. Alansary, "A Practical Application of the UNL+3 Program on the Arabic Language", in *Proceedings of the 10th Egyptian Society of Language Engineering Conference*, Ain Shams University, Cairo, Egypt, December 15 – 16, 2010.
- [22] S. Alansary, M. Nagi and N. Adly, "A Semantic-Based Approach for Multilingual Translation of Massive Documents", in *Proceedings of The 7th International Symposium on Natural Language Processing*, (SNLP), Pattaya, Thailand, 2007.
- [23] S. Alansary, M. Nagi and N. Adly, "The Universal Networking Language in Action in English-Arabic Machine Translation", in *Proceedings of 9th Egyptian Society of Language Engineering Conference on Language Engineering*, (ESOLEC 2009), Cairo, Egypt December 23-24, 2009.
- [24] N. Adly, and S. Alansary, "Evaluation of Arabic Machine Translation System based on the Universal Networking Language", in *Proceedings of the 14th International Conference on Applications of Natural Language to Information Systems*, (NLDB 2009), Saarland University, Saarbrücken-Germany, June 24 - 26 2009.
- [25] S. Alansary, M. Nagi and N. Adly, "A Library Information System (LIS) Based on UNL Knowledge Infrastructure". in *Proceedings of the Universal Networking Language Workshop In conjunction with 7th International Conference on "computer science and information technology - 2009"* (CSIT-2009), Yerevan – Armenia, September 28th - October 2nd , 2009.

Mining Opinion in Arabic Data: A Comparison between Supervised and Unsupervised Classification Approaches

Ahmed M. Misbah^{*1}, Ibrahim F. Imam^{*2}

**Computer Science Department, Faculty of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport*

2033 - El Horreya, El Moshir Ismail St., behind Sheraton Building, Cairo, Egypt

¹ahmed.misbah@hotmail.com

²ifi05@yahoo.com

Abstract— Opinion Mining can be described as the task of detecting subjectivity in a given text and measuring its polarity. Many research papers have presented experimentation carried out in different domains in the English language such as movie reviews, political forums and blogs. Work on the Arabic language has been very limited due to the lack of Arabic content on the World Wide Web written in non slang, classical Arabic. In this paper, a number of supervised and unsupervised learning algorithms used for Opinion Mining were trained and tested on Arabic Religious decrees. Choosing this domain was due to the fact that religious decrees are written in classical Arabic. Best results were obtained using Support Vector Machine algorithm giving an accuracy rate of 79%.

1 INTRODUCTION

Textual information in the world can be categorized into two main types: facts and opinions. Facts are objective expressions about entities, events and their properties. Opinions are subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties.

Opinion Mining aims to detect subjective expressions in text and measure the polarity of sentiment and feelings. It is also expressed in other terms such as Sentiment Analysis and Subjectivity Analysis. It is an area in Text Mining that was promoted by the widespread of user generated content on the World Wide Web [1]. Web Applications based on user generated content contain large amounts of text expressing opinions, reviews, and critics on different products and events. Examples of such web applications are Web Blogs, Internet Forums, discussion groups, and review sites such as Blogger, Epinions.com, CNET and Amazon.

Opinion Mining on Arabic text is not popular among researches due to the lack of good quality data. This paper presents an approach (explained in Section 3) using a number of supervised and unsupervised learning algorithms on a unique Arabic dataset. A large dataset of Arabic Religious Decrees was used to carry out the experimentation. The learning algorithms' accuracies were measured based on how accurate the classification of the Religious Decrees to Halal (Allowed) and Haraam (Prohibited) polarities was.

2 OPINION MINING AND SENTIMENT ANALYSIS

Prior to the year 2001, very few researches addressed the problems and challenges Opinion Mining and Sentiment Analysis raised. Such researches only focused on interpretation of metaphor, narrative, point of view, effect, and related areas [2]-[5].

The year 2001 marked the beginning of the widespread of awareness of research problems and opportunities Opinion Mining and Sentiment Analysis raised. Since then, hundreds of research papers were published in this area.

A number of reasons are believed to have promoted this area of research:

- The rise of Machine Learning methods in Natural Language Processing and Information Retrieval.
- The availability of datasets for Machine Learning algorithms to be trained on due to huge widespread of review related sites on the World Wide Web.
- Realization of opportunities this field would offer in developing commercial applications.

The term "Opinion Mining" appears in a paper by Dave et al. [6]. According to this paper, the ideal opinion-mining tool would "process a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good)." Much of their subsequent research on opinion mining fits this description in its emphasis on extracting and analyzing judgments on various aspects of given items. However, the term Opinion Mining has recently been interpreted more broadly to include many different types of text analysis.

The term “Sentiment Analysis” appears within the same time frame of the term “Opinion Mining”. The term “sentiment” used in reference to the automated analysis of text and tracking of the predictive judgments appears in 2001 papers by Das et al. [7] and Tong [8]. That was due to the authors’ interest in analyzing market sentiment. It subsequently occurred within 2002 papers by Turney [9] and Pang et al. [10].

A number of papers mentioning “Sentiment Analysis” focus on the classification of reviews based on their polarities (either positive or negative) [11], [12]. This fact appears to have caused some authors to suggest that the phrase refers specifically to this narrowly defined task. However, recent researchers interpret the term more broadly to mean the computational treatment of opinion, sentiment, and subjectivity in text. Thus, when broad interpretations are applied, “Sentiment analysis” and “Opinion mining” denote the same field of study.

Very few research papers address Opinion Mining and Sentiment Analysis in languages other than English [13]-[18]. Those research papers address the field in multiple languages such as Chinese, Urdu and Arabic. The fact that very little work exists in Multilingual Opinion Mining indicates the lack of multilingual corpora for benchmarking newly developed systems and approaches. It also indicates that there might be language related issues in the field that have not yet been explored.

3 EXPERIMENTATION

A. Data Collection

The data collected for conducting the experimentation were Arabic Religious Decrees. The reason behind choosing this particular category is that Religious decrees are known to express sentiment and contain subjective text. Another reason was the scarcity of Arabic subjective text that expresses opinion in other domains, such as reviews, written in classical Arabic and not in any Arabic slang.

Data was collected from 5 well known and acknowledged Islamic sites:

1. Islam Way (www.islamway.com)
2. Islam Online (www.islamonline.net)
3. Islam QA (islamqa.com/ar)
4. Islam Web (www.islamweb.net)
5. Al Eman (www.al-eman.com/)

The total amount of decrees downloaded was 77,047.

B. Simple Text Preprocessing

Simple Text preprocessing was executed against the data crawled from the web to prepare it for manual labeling. This simple text preprocessing included:

1. Removing HTML Tags
2. Removing Non Arabic characters
3. Removing special characters

Figures 1 and 2 illustrate an HTML file before and after simple text preprocessing:

```

<html DIR=LTR>
<head>
<title>الفتوى بين يديك</title>
<td width="54%" align="right" bgcolor="#ffffff" dir="rtl"><font face="Simplified Arabic" size=3><b>
فقه العبادات الموضوع الرئيسي اختيار جمهور العلماء في فوائت الصلاة عنوان الفتوى أبو عبد الله محمد البخاري اسم المفتى رقم
المفتى </b></font></td><td width="100%" dir="RTL" align="right" colspan="3"
bgcolor="#ffffff">&nbsp;
<font face="Simplified Arabic" size=3><b>السلام عليكم ورحمة الله وبركاته</b><BR>
ماذا أفعل في الصلاة التي لم أصلها؟
<p>&nbsp;</p></td>
<td width="100%" align="right" colspan="3" dir="RTL" bgcolor="#ffffff"><font face="Simplified Arabic"
size="3"> <b><P class=DetailFont align=right><FONT color=#000000 size=4>الحمد لله والصلاة والسلام على رسول الله</FONT>
فإن ترك الصلاة ذنب عظيم، وكبيرة من أكبر الكبائر، بل قد صرحت الأحاديث بكفر <BR>
تاركها، كما روى مسلم في صحيحه عن جابر رضي الله عنه قال: سمعت رسول الله صلى الله عليه وسلم يقول: (بين الرجل وبين
والواجب على العبد - كما هو اختيار جمهور العلماء وهو الذي نرجحه - أن يبادر إلى <BR>
قضاء ما فاتته من صلوات على الفور، بحسب استطاعته، وسواء كان ذلك ليلاً أو نهاراً، مع مراعاة الترتيب بين
الواجب على العبد كما هو اختيار جمهور العلماء وهو الذي نرجحه أن يبادر إلى قضاء ما فاتته من صلوات على الفور بحسب استطاعته وسواء كان ذلك ليلاً أو
نهاراً مع مراعاة الترتيب بين الفوائت. والله أعلم.</FONT></P></b>&nbsp;</td>
</tr>
</table><a href="javascript:back()" >الصفحة السابقة</a>

```

Figure 1: HTML file before Simple Text Preprocessing

الفتوى بين يديك فقه العبادات الموضوع الرئيسي اختيار جمهور العلماء في فوائت الصلاة عنوان الفتوى أبو عبد الله محمد البخاري اسم المفتى رقم الفتوى تاريخ الفتوى على الموقع نص السؤال السلام عليكم ورحمة الله وبركاته ماذا أفعل في الصلاة التي لم أصلها نص الفتوى الحمد لله والصلاة والسلام على رسول الله وعلى آله وصحبه أما بعد فإن ترك الصلاة ذنب عظيم وكبيرة من أكبر الكبائر بل قد صرحت الأحاديث بكفر تاركها كما روى مسلم في صحيحه عن جابر رضي الله عنه قال سمعت رسول الله صلى الله عليه وسلم يقول بين الرجل وبين الكفر ترك الصلاة. والواجب على العبد كما هو اختيار جمهور العلماء وهو الذي نرجحه أن يبادر إلى قضاء ما فاتته من صلوات على الفور بحسب استطاعته وسواء كان ذلك ليلاً أو نهاراً مع مراعاة الترتيب بين الفوائت. والله أعلم. الصفحة السابقة

Figure 2: HTML file after Simple Text Preprocessing

C. Manual Data Labeling

The data collected from Islamic sites lacked labels to indicate its polarity (Halal or Haraam). It was required to manually label the data and insert it into a database in order to be able to measure the text's polarities [21].

A Java desktop application was created to label the data into 4 different categories:

1. Halal (Decrees that clearly indicate that the topic inquired for is allowed)
2. Haraam (Decrees that clearly indicate that the topic inquired for is prohibited)
3. Both (Decrees that contain both opinions Halal and Haraam)
4. None (Decrees that are strictly objective and do not contain any opinion or subjective text)

The desktop application also splits the data into a question and answer in order to mine for opinion only within the answers.

Figures 3 and 4 illustrate a data file before and after manual labeling:

الفتوى بين يديك فقه العبادات الموضوع الرئيسي اختيار جمهور العلماء في فوائت الصلاة عنوان الفتوى أبو عبد الله محمد البخاري اسم المفتى رقم الفتوى تاريخ الفتوى على الموقع نص السؤال السلام عليكم ورحمة الله وبركاته ماذا أفعل في الصلاة التي لم أصلها نص الفتوى الحمد لله والصلاة والسلام على رسول الله وعلى آله وصحبه أما بعد فإن ترك الصلاة ذنب عظيم وكبيرة من أكبر الكبائر بل قد صرحت الأحاديث بكفر تاركها كما روى مسلم في صحيحه عن جابر رضي الله عنه قال سمعت رسول الله صلى الله عليه وسلم يقول بين الرجل وبين الشرك والكفر ترك الصلاة. والواجب على العبد كما هو اختيار جمهور العلماء وهو الذي نرجحه أن يبادر إلى قضاء ما فاتته من صلوات على الفور بحسب استطاعته وسواء كان ذلك ليلاً أو نهاراً مع مراعاة الترتيب بين الفوائت. والله أعلم. الصفحة السابقة

Figure 3: Text before manual labelling

Question: السلام عليكم ورحمة الله وبركاته ماذا أفعل في الصلاة التي لم أصليها
الحمد لله والصلاة والسلام على رسول الله وعلى آله وصحبه أما بعد فإن ترك الصلاة ذنب عظيم وكبيرة من أكبر الكبائر بل قد صرحت
الأحاديث بكفر تاركها كما روى مسلم في صحيحه عن جابر رضي الله عنه قال سمعت رسول الله صلى الله عليه وسلم يقول بين الرجل وبين الشرك والكفر
ترك الصلاة. والواجب على العبد كما هو اختيار جمهور العلماء وهو الذي نرجحه أن يبادر إلى قضاء ما فاتته من صلوات على الفور بحسب استطاعته
وسواء كان ذلك ليلاً أو نهاراً مع مراعاة الترتيب بين الفوائت
Issuer: ISSUER

Figure 4: Text after manual labelling

Table I illustrates the number of files in each category. It is observed that the total amount of data has decreased from the original number collected. That is due to the fact that a number of files were labeled corrupt due to redundancy or irrelevant data.

TABLE I
DECREE COUNT IN EVERY CATEGORY

Category	Count
Halal	8689
Haram	10355
Both	8455
None	34064

D. Advanced Text Preprocessing

More advanced Text Preprocessing was executed against the data to prepare it for input into different learning algorithms. It was done only the answers obtained from the religious decrees.

The first step that was executed was the removal of Arabic stop words from the text except negation letters as they tend to shift the polarity of a given term. The following example explains this fact:

Arabic Sentence: ليس جيد

English Sentence: Not Good

The word جيد has a positive polarity. But when preceded by the word ليس, which is a negation letter, its polarity is shifted to the opposite making the sentence express negative opinion.

The list of Arabic stop words was gathered from a project on Source Forge (<http://sourceforge.net/projects/arabicstopwords>).

The list of Arabic negation letters that were excluded from Arabic stop words included:

- ليس
- غير
- لم
- لَمَّا
- لَنْ
- ما
- لا
- لات

The second step that was executed was Part of Speech Tagging. The tool that was used to determine the POS tags of tokens was The Stanford Log-Linear Part of Speech Tagger (<http://nlp.stanford.edu/software/tagger.shtml>).

The third step that was executed was Feature Extraction. Feature Extraction involves extracting tokens that are relevant to detecting sentiment and measuring polarity in the document. The following features were extracted from documents:

- Adjectives (with and without negation letters)
- Adverbs (with and without negation letters)
- Nouns (with and without negation letters)

- Verbs (with and without negation letters)

Finally the tokens extracted after feature extraction are stemmed using Buckwalter Arabic Stemmer.

E. Weight Calculation

In order to calculate the polarity of the features extracted from Text Preprocessing, a certain weight was required to be calculated for each feature. Semantic Orientation using Pointwise Mutual Information was used to calculate the weights. The original SO-PMI equation derived by P.Turney in 2002 [9] was as follows:

$$SO - PMI(\text{word}) = \log_2 \frac{\text{hits}(\text{word NEAR } p_{\text{query}}) \text{ hits}(n_{\text{query}})}{\text{hits}(\text{word NEAR } n_{\text{query}}) \text{ hits}(p_{\text{query}})} \quad (1)$$

However this equation is only efficient for the English language. A modified version of the equation by G.Wang et. al. in 2008 [19] proved more efficient for other languages. Therefore the equation used in my experimentation was as follows:

$$SO - PMI(\text{word}) = \log_2 \frac{\text{hits}(\text{word AND } p_{\text{query}}) \text{ hits}(n_{\text{query}})}{\text{hits}(\text{word AND } n_{\text{query}}) \text{ hits}(p_{\text{query}})} \quad (2)$$

The values of pquery and nquery were:

P_QUERY=حلال OR يجوز OR مباح OR مستحب OR مشروع OR يصح OR سنة OR فرض OR واجب

N_QUERY=حرام OR مكروه OR ممنوع OR خطأ OR باطل OR منع OR بدعة OR فاسد OR مذموم

The Yahoo search engine was used to calculate the number of hits of each token in relation with pquery and nquery. The reason why Yahoo was selected was that it allows for an unlimited number of queries unlike Google which limits that number.

Weights were calculated for both Stemmed and Non Stemmed tokens. The reason why this was done was the assumption that Non Stemmed tokens would return more accurate weights than Stemmed tokens. The assumption was made due to the fact that tokens usually used in documents that will be retrieved from a web based search engine will not be stemmed.

F. Experimentation using Unsupervised Learning Algorithm

After all Text Preprocessing was accomplished and the weights for each feature in the feature vector were calculated, experimentation was conducted on an Unsupervised Learning Algorithm proposed by P.Turney in 2002 [9]. In this paper, the algorithm will be named Average SO-PMI.

Average SO-PMI calculates the average weights in every document's feature vector. But unlike Turney's approach, the term count of every feature is multiplied by the weight value before the average is calculated. The value obtained is then compared with a threshold. If the value is greater than the threshold, then the document expresses positive opinion, otherwise the document expresses negative opinion.

Different feature vectors for every document were used in the experimentation. Figures 5-8 demonstrate the results obtained from this approach using 7 threshold values and 4 feature vectors.

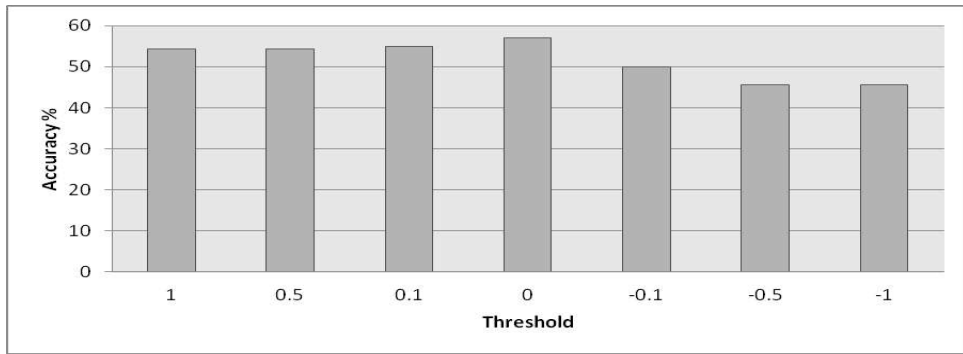


Figure 5: Results obtained using All unigrams+bigrams(valence shifters)+POS+Stems

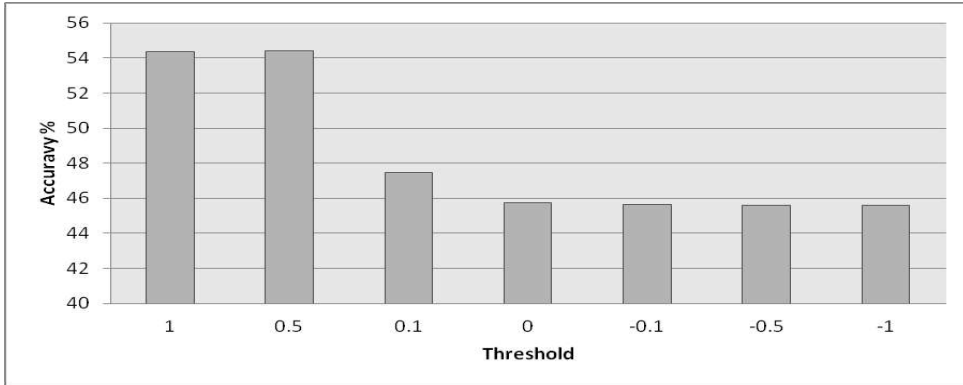


Figure 6: Results obtained using All unigrams+bigrams(valence shifters)+POS+Non Stems

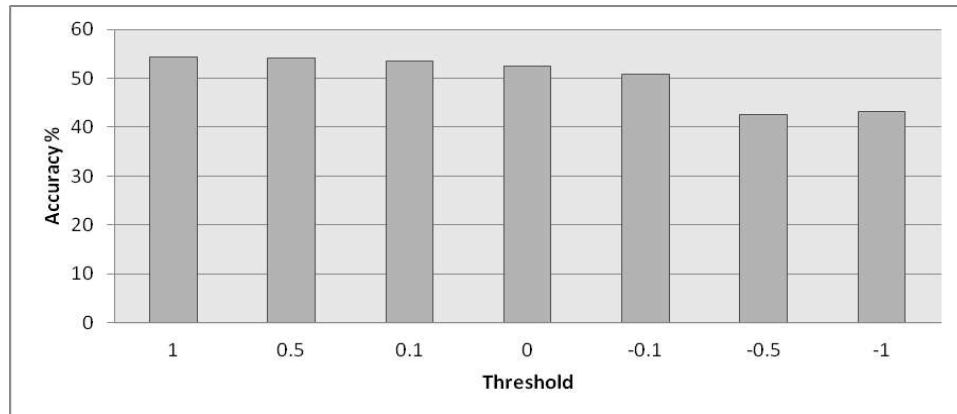


Figure 7: Results obtained using Adjective/Adverb unigrams+bigrams(valence shifters)+POS+Stems

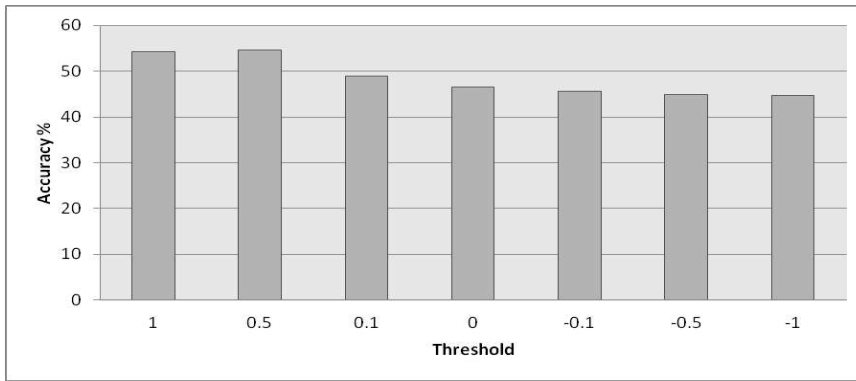


Figure 8: Results obtained using Adjective/Adverb unigrams+bigrams(valence shifters)+POS+Non Stems

G. Experimentation using Supervised Learning Algorithms

1) *Support Vector Machine Classifier*: Support Vector Machine (SVM) has been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes. They are large-margin, rather than probabilistic, classifiers, in contrast to Naive Bayes and Maximum Entropy. In the two-category case, the basic idea behind the training procedure is to find a hyperplane, represented by vector $\vec{\omega}$, that not only separates the document vectors in one class from those in the other, but for which the separation, or margin, is as large as possible. This search corresponds to a constrained optimization problem; letting $c_j \in \{1, -1\}$ (corresponding to positive and negative) be the correct class of document d_j , the solution can be written as:

$$\vec{\omega} = \sum_1 \alpha_j c_j \vec{d}_j, \quad \alpha_j \geq 0 \quad (3)$$

where the α_j are obtained by solving a dual optimization problem. Those \vec{d}_j such that α_j is greater than zero are called support vectors, since they are the only document vectors contributing to $\vec{\omega}$. Classification of test instances consists simply of determining which side of $\vec{\omega}$'s hyperplane they fall on [10].

SVM Light library (<http://svmlight.joachims.org>) was used in this experimentation. The experimentation was executed using 2 Cross-Fold-validation, two weighting schemas (SO-PMI and Presence) and 4 feature vectors. SO-PMI weight for each term in the vector is multiplied by the word count in the document. Presence term vector puts a 1 or 0 value as the weight of the term depending on its occurrence in the document. If the term is present in the document a 1 is used otherwise 0. Term count in this weighting schema is ignored [20].

2) *Naive Bayes Classifier*: A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". It assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of any other feature.

Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naive Bayes models uses the method of maximum likelihood; in other words, one can work with the Naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

In Text Classification, a Naive Bayes classifier assigns a given document d the class:

$$c^* = \operatorname{argmax} P(c | d) \quad (4)$$

By first observing Bayes' rule:

$$P(c | d) = \frac{P(c) P(d | c)}{P(d)} \quad (5)$$

The equation for obtaining PNB(c | d) is derived where P(d) plays no role in selecting c*. To estimate the term P(d | c), Naive Bayes decomposes it by assuming the fi's are conditionally independent given d's class [22]:

$$P_{NB}(c | d) = \frac{P(c) (\prod_{i=1}^m P(f_i | c))^{n_i(d)}}{P(d)} \quad (6)$$

LingPipe Library (<http://alias-i.com/lingpipe/index.html>) was used in this experimentation. The experimentation was executed using 2 Cross-Fold-validation and 4 feature vectors. Term weighting was neglected in this approach to prevent running a discretization tool to obtain weight values using a Gaussian distribution assumption.

3) *K-Nearest Neighbor Classifier*: K-Nearest Neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor [22].

kNN algorithm can be used in Subjectivity Classification where text is represented in the Vector Space Model and the distance between the class's centroid and incoming document vector is measured using distance metric such as Euclidean distance. LingPipe Library was used for this experimentation. And similar to Naïve Bayes, Presence was used for weighting and not SO-PMI. The value of k used was equal to 5 and Euclidean distance was used as a distance metric.

4) *Supervised Learning Algorithms' Results*: Figures 9-12 demonstrate the accuracies obtained using Supervised Learning Algorithms.

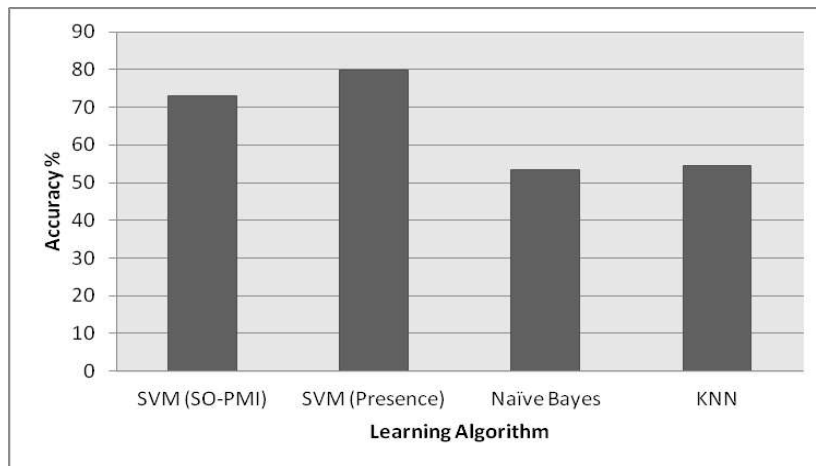


Figure 9: Results obtained using All unigrams+bigrams(valence shifters)+POS+Stems

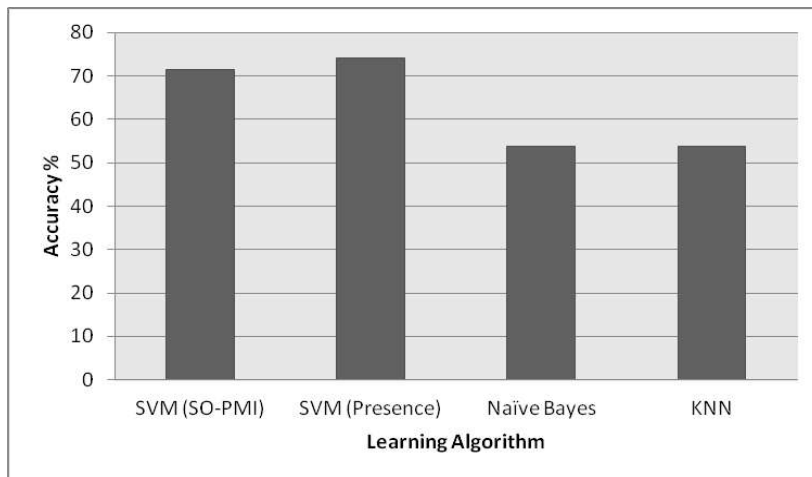


Figure 10: Results obtained using All unigrams+bigrams(valence shifters)+POS+Non Stems

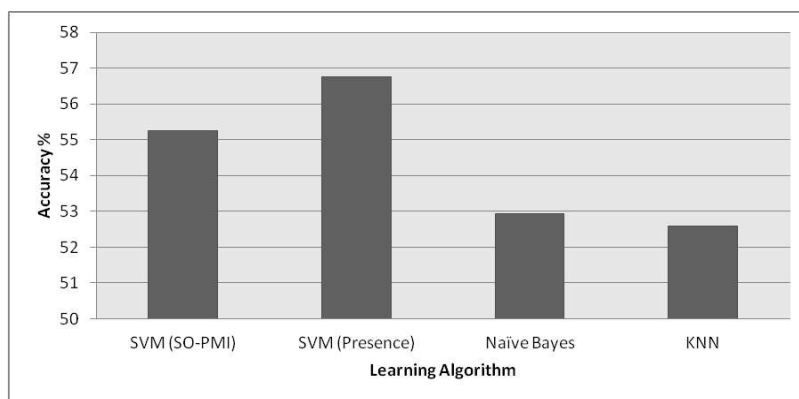


Figure 11: Results obtained using Adjective/Adverb unigrams+bigrams(valence shifters)+POS+Stems

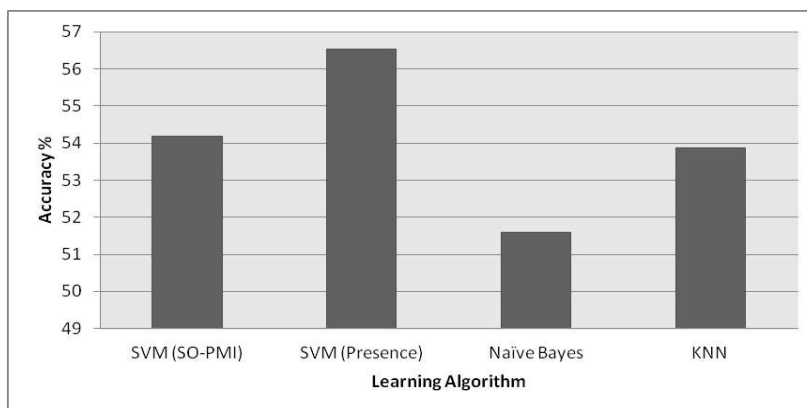


Figure 12: Results obtained using Adjective/Adverb unigrams+bigrams(valence shifters)+POS+Non Stems

4 CONCLUSIONS

It was concluded from the results that the best accuracy rate was obtained by using Support Vector Machine utilizing all stemmed unigrams, bigrams (negation letters) and presence as a weighing schema.

SO-PMI values were inaccurate due to the fact that web based search engines contain poor Arabic data. The inaccurate SO-PMI values affected Average SO-PMI algorithm as it relies mainly on calculating the average of SO-PMI weights of the document's term vector.

Using all features performed better than using only adjective and adverb features unlike what was achieved in experimentation done on English language. It was expected that adjectives and adverbs would enhance the quality of the classification since this type of POS is always subjective. That was due to the fact that the SO-PMI values calculated for those POSs were inaccurate and that the size of the term vector was reduced considerably by using less features.

Naïve Bayes and kNN performed very poorly when compared to SVM. Neglecting weights and the reliance on presence could have been a cause. For kNN, the use of Euclidian was not enough and it should have been tested using other distance metrics such as Cosine similarity which is known to perform well with kNN.

It is recommended that future calculations of SO-PMI be executed on a corpus other than Yahoo's or Google's. This would guarantee the quality of the Arabic data on which the weights are calculated.

REFERENCES

- [1] Bing Liu, *Sentiment Analysis and Subjectivity, Handbook of Natural Language Processing*, Second Edition, (editors: N. Indurkha and F. J. Damerau), 2010.
- [2] M. Kantrowitz, "Method and apparatus for analyzing affect and emotion in text," U.S. Patent 6622140, Patent filed in November 2000, 2003.
- [3] J. Wiebe and R. Bruce, "Probabilistic classifiers for tracking point of view," in *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pp. 181–187, 1995.
- [4] J. M. Wiebe, "Tracking point of view in narrative," in *Computational Linguistics*, vol. 20, pp. 233–287, 1994.
- [5] J. M. Wiebe, R. F. Bruce, and T. P. O'Hara, "Development and use of a gold standard data set for subjectivity classifications," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 246–253, 1999.
- [6] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of WWW*, pp. 519–528, 2003.
- [7] S. Das and M. Chen, "Yahoo! for Amazon: Extracting market sentiment from stock message boards," in *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*, 2001.
- [8] R. M. Tong, "An operational system for detecting and tracking opinions in on-line discussion," in *Proceedings of the Workshop on Operational Text Classification (OTC)*, 2001.
- [9] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 417–424, 2002.
- [10] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86, 2002.
- [11] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the Conference on Knowledge Capture (K-CAP)*, 2003.
- [12] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2003.
- [13] A Abbasi, H Chen, A Salem, "Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums," in *ACM Transactions on Information Systems (TOIS)*, Volume 26, Issue 3, Article #12, 2008.
- [14] K Ahmad, D Cheng, Y Almas, "Multi-lingual Sentiment Analysis of Financial News Streams," in *Proceedings of Science, Grid Technology for Financial Modeling and Simulation*, Italy, 2006.
- [15] R. Mihalcea, C. Banea, and J. Wiebe, "Learning multilingual subjective language via cross-lingual projections," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 976–983, Prague, Czech Republic, June 2007.
- [16] Y Almas, K Ahmad, "A note on extracting 'sentiments' in financial news in English, Arabic & Urdu," in *Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages Linguistic Institute*, Stanford, California, USA, pp. 1-12, 2007.
- [17] M Elhawary, M Elfeky, "Mining Arabic Business Reviews," in *IEEE International Conference on Data Mining Workshops*, 2010.
- [18] N Farra, E Challita, R Abou Assi, H Hajj, "Sentence-Level and Document-Level Sentiment Mining for Arabic Texts," in *IEEE International Conference on Data Mining Workshops*, 2010.
- [19] G Wang, K Araki, "An unsupervised opinion mining approach for Japanese weblog reputation information using an improved SO-PMI algorithm," in *IEICE Transactions on Information and Systems*, ISSN 0916-8532, Vol. 91, N° 4 , pages 1032-1041, 2008
- [20] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," in *Computational Intelligence*, vol. 22, pp. 110–125, 2006.
- [21] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 115–124, 2005.
- [22] Mohammed J. Bawaneh, Mahmud S. Alkoffash and Adnan I. Al Rabea, "Arabic Text Classification using K-NN and Naive Bayes," in *Journal of Computer Science 4 (7)*: pp. 600-605, 2008

Automatic Speech Annotation Using HMM based on Best Tree Encoding (BTE) Feature

Amr M. God^{*1}, Rania Ahmed Abul Seoud^{*2}, Mohamed Hassan^{*3}

**Electrical Engineering, Faculty of Engineering, Fayoum University
Egypt*

¹ amg00@fayoum.edu.eg

² r-abulseoud@k-space.org

³ mh1323@fayoum.edu.eg

Abstract— Manual annotation for time-aligning a speech waveform against the corresponding phonetic sequence is a tedious and time consuming task. This paper aimed to introduce a completely automated phone recognition system based on Best Tree Encoding (BTE) 4-point speech feature. BTE is used to find phoneme boundaries along speech utterance. Comparison to Mel-frequency cepstral coefficients (MFCCs) speech feature in solving the same problem is provided. Hidden Markov Model (HMM) and Gaussian Mixtures are used for building the statistical models through this research. HTK software toolkit is utilized for implementation of the model. The System can identify spoken phone at 65.1% recognition rate based on MFCC and 57.2% recognition rate based on BTE. The current BTE vector is 4 components compared to 39 components of MFCC. This makes it very promising features vector, BTE with 4 components gives a comparable recognition success rate compared to the 39 components MFCC vector widely in the area of ASR.

Keywords—BTE, MFCC, HTK, Gaussian Mixture, speech recognition

1 INTRODUCTION

Presently, manual annotation by expert phoneticians is the most precise way for time-aligning a speech waveform against the corresponding phonetic sequence. This is a tedious and time consuming task, which makes it a prohibitive choice for large speech corpora. Several approaches have been proposed for the task of speech segmentation [2-6]. The most frequently used approach is based on HMM phone models. In this method each speech waveform is initially decomposed into a sequence of feature vectors, using a speech parameterization technique. Afterwards, a set of HMM phone models (phone recognizer) is utilized to extract the corresponding phonetic sequence as well as the positions of the phonetic boundaries. Other speech segmentation methods have also been proposed in the literature. Some of them include detection of variations/similarities in spectral or prosodic parameters of speech, template matching using dynamic programming and/or synthetic speech and discriminative learning segmentation.

Various speech parameterizations have been utilized in the phonetic segmentation task, with the Mel Frequency Cepstral Coefficients (MFCC) among the most widely used, especially in the HMM-based approach. Other speech features such as Perceptual Linear Prediction (PLP), Line Spectral Frequencies (LSF), Linear Predictive Coding (LPC), short-time energy, formants and wavelet-based have also been used.

Automatic annotation is used to make a preliminary solution before starting the manual annotation. Its task is to simplify the effort in the manual annotation task. In this paper, the most frequently approach – adapting a Hidden Markov Model (HMM) based phonetic recognizer to the task of automatic phonetic segmentation is used. Our base line system contains 10ms frame rate with 25ms Hamming window. Here the speech is parameterized using MFCC and BTE. MFCC with 12 Mel-Frequency Cepstral Coefficients and normalized log energy, as well as their first and second order differences yielding a total of 39 components. Another parameterization technique is Best Tree Encoding BTE with 4 spectral based components. A set of context-independent Left -To -Right (LR) monophone HMMs with one Gaussian per state are flat-initialized. The HMM model is 3 emitting states. These HMMs are well trained using the HMM Tool Kit (HTK¹) and both features MFCC and BTE for the problem of automatic annotation.

Speech database is prepared to measure the quality of this experiment. Speech database is labeled and transcribed then verified to evaluate the results of automatic segmentation. The following sections will navigate through the details of this research. Section 2 will illustrate problem definition. In section 2, the HMM GMM based speech recognition will be illustrated.

¹ HTK is available through the following URL <http://htk.eng.cam.ac.uk/>. University of Cambridge.

BTE speech feature is explored in section 3. The experimental Framework will be provided in section 4. The experimental procedure will be presented in section 5. The results will be presented in section 6. The conclusion will be given in section 7. Then finally the list of references will be listed in section 8.

2 PROBLEM DEFINITION

Automatic Speech annotation to Arabic phone level is the problem that is intended in this research. The phone is supposed to be the basic speech unit. Finding the phone boundaries along the stream of human speech is the basic definition of the annotation. Speech features should be stable along the phone duration. The best the features are the accurate the boundaries are.

3 HMM–GMM BASED SPEECH RECOGNITION

In HMM–GMM (Hidden Markov Model –Gaussian Mixture model related) based speech recognition ,see Gales and Young, 2007 for review[10], the short-time spectral Characteristics of speech is turned into a vector (the “observations” of Fig. 1, sometimes called frames), and build a generative model using a HMM that produces sequences of these vectors. A left-to-right three-state HMM topology as in Fig. 1 will typically model the sequence of frames generated by a single phone. Models for sentences are constructed by concatenating HMMs for sequences of phones. Different HMMs are used for phones in different left and right phonetic contexts, using a tree-based clustering approach to model unseen contexts ,see Young et al., 1994 for review [11]. the index j will be used for the individual context-dependent phonetic states, with $1 \leq j \leq J$. While j could potentially equal three times the cube of the number of phones (assuming only the immediate left and right phonetic context will be modelled), after tree-based clustering it will typically be several thousand. The distribution that generates a vector within HMM state j is a Gaussian Mixture Model (GMM):

$$P(x|j) = \sum_{i=1}^{M_j} w_{ji} N(x, \mu_{ji}, \Sigma_{ji}) \quad (1)$$

Table 1 shows the parameters of the probability density functions (pdfs) in an example system of this kind: each context dependent state (of which we only show three rather than several thousands) has a different number of sub-states M_j .

TABLE 1
PARAMETERS FOR PDFS IN GMM HMM SYSTEM

State 1	State 2	State 3
$\mu_{11}, \Sigma_{11}, w_{11}$	$\mu_{21}, \Sigma_{21}, w_{21}$	$\mu_{31}, \Sigma_{31}, w_{31}$
$\mu_{12}, \Sigma_{12}, w_{12}$	$\mu_{22}, \Sigma_{22}, w_{22}$	$\mu_{32}, \Sigma_{32}, w_{32}$
$\mu_{13}, \Sigma_{13}, w_{13}$	$\mu_{23}, \Sigma_{23}, w_{23}$	
	$\mu_{24}, \Sigma_{24}, w_{24}$	

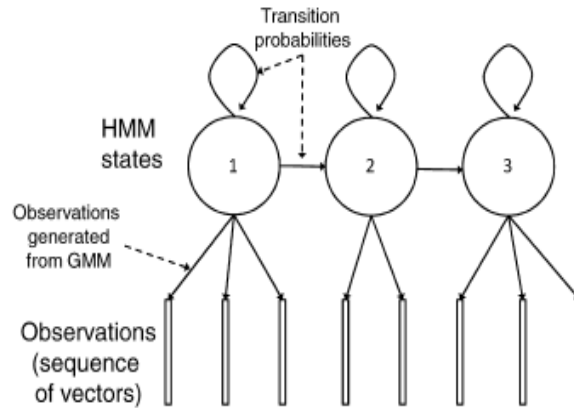


Figure1: HMM for speech recognition

HTK is principally concerned with continuous density models in which each observation probability distribution is represented by a mixture Gaussian density. In this case, for state j the probability $b_j(o_t)$ of generating observation o_t is given by

$$b_j(o_t) = \prod_{s=1}^S \left[\sum_{m=1}^{M_{js}} c_{j sm} N(o_{st} >; \mu_{sm}, \Sigma_{j sm}) \right]^{r_s} \quad (2)$$

where M_{js} is the number of mixture components in state j for stream s , c_{jSm} is the weight of the m^{th} component and $N(\mathbf{o}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, that is

$$N(\mathbf{o}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{o}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{o}-\boldsymbol{\mu})} \quad (3)$$

where n is the dimensionality of \mathbf{o} . The exponent is a stream weight and its default value is one. Other values can be used to emphasise particular streams, however, none of the standard HTK tools manipulate it. HTK also supports discrete probability distributions in which case

$$b_j(\mathbf{o}_t) = \prod_{s=1}^S \{P_{js}[v_s(\mathbf{o}_{st})]\} \quad (4)$$

Where $v_s(\mathbf{o}_{st})$ is the output of the vector quantiser for stream s given input vector \mathbf{o}_{st} and $P_{js}[v]$ is the probability of state j generating symbol v in stream s . In addition to the above, any model or state can have an associated vector of duration parameters $\{d_k\}_1$. Also, it is necessary to specify the kind of the observation vectors, and the width of the observation vector in each stream. Thus, the total information needed to define a single HMM are listed as follows

- Type of observation vector
- Number and width of each data stream
- Optional model duration parameter vector
- Number of states
- For each emitting state and each stream
 - mixture component weights or discrete probabilities
 - if continuous density, then means and covariance
 - optional stream weight vector
 - optional duration parameter vector
- Transition matrix

In Automatic Speech Recognition (ASR) system, it is normally used Gaussian mixture HMMs as acoustic models for modeling basic speech units, ranging from context-independent whole words in small vocabulary ASR tasks to context-dependent phonemes (e.g., triphones) in large vocabulary ASR. Traditionally, the HMM-based acoustic models are estimated from available training data using the well-known EM algorithm based on the maximum-likelihood (ML) criterion. To deal with data sparseness problems in model training, we normally use phonetic decision trees to tie HMM states from different triphone contexts. In order to derive a simple closed-form solution, we normally grow the decision trees based on simple models, such as single Gaussian HMMs. After the state-tied structure is determined from the decision trees, a separate “mixing-up” step is used to gradually increase the number of Gaussian mixtures in each tied HMM state until the optimal performance is achieved. In today’s ASR systems, e.g., HTK, “mixing-up” is normally implemented in two steps [2]:

- 1) All existing Gaussians or the most dominant Gaussian mixture component in an HMM state is split based on some random or heuristic strategies.
- 2) All split Gaussians are re-estimated based on the EM algorithm.

Obviously, this incremental method for increasing model complexity is a good strategy to learn very large-scale statistical models without getting trapped in any bad local optimum. However, we still face some problems when increasing model complexity in the above “mixing-up” strategy. First of all, the random splitting strategy is not optimal in terms of the model estimation criterion. For example, there is no guarantee that the newly added Gaussian components from random splitting always increase the likelihood function prior to re-estimation. Second, since the subsequent EM-based re-estimation is sensitive to the initial parameters of the randomly split Gaussians, there is no guarantee that the EM-based re-estimation can always converge to the optimal point.

In HTK, the conversion from single Gaussian HMMs to multiple mixture component HMMs is usually one of the final steps in building a system. The mechanism provided to do this is the HHED MU command which will increase the number of components in a mixture by a process called *mixture splitting*. This approach of building a multiple mixture component system is extremely flexible since it allows the number of mixture components to be repeatedly increased until the desired level of performance is achieved. The MU command has the form

MU n itemList

Where n gives the new number of mixture components required and `itemList` defines the actual mixture distributions to modify. This command works by repeatedly splitting the mixture with the largest mixture weight until the required number of components is obtained. The actual split is performed by copying the mixture, dividing the weights of both copies by 2, and finally perturbing the means by plus or minus 0.2 standard deviations. For example, the command has the form

MU n itemList

For example, the command

MU 3 {aa.state[2].mix}

would increase the number of mixture components in the output distribution for state 2 of model `aa` to 3. Normally, however, the number of components in all mixture distributions will be increased at the same time. Hence, a command of the form is more usual

MU 3 {*.state[2-4].mix}

It is usually a good idea to increment mixture components in stages, for example, by incrementing by 1 or 2 then re-estimating, then incrementing by 1 or 2 again and re-estimating, and so on until the required number of components is obtained. This also allows recognition performance to be monitored to find the optimum.

We can start prototype of phone in HMM with 4 mixtures per state. However, this was (a pretty good) guess of us. To be sure that we have chosen the optimal topology for the models there is no way to avoid the heuristic try-and-fail method. We ran a series of trainings on different number of mixtures. It is recommended to start with a single Gaussian model, train it until it converges on the dev set and then increase the number of mixtures by one, train them and so on.

One final point with regard to multiple mixture component distributions is that all HTK tools ignore mixture components whose weights fall below a threshold value called `MINMIX` (defined in `HModel.h`). Such mixture components are called *defunct*. Defunct mixture components can be prevented by setting the `-w` option in `HEREST` so that all mixture weights are floored to some level above `MINMIX`. If mixture weights are allowed to fall below `MINMIX` then the corresponding Gaussian parameters will not be written out when the model containing that component is saved. It is possible to recover from this, however, since the `MU` command will replace defunct mixtures before performing any requested mixture component increment.

4 BEST TREE ENCODING

BTE is a simple on/off entropy mapping of the signal into the bands in which the signal is decomposed using wavelet packets. The key property in BTE is the alignment of the neighboring frequency domain bands in wavelet packets decomposition of the signal. Adjacent bands are much closer in distance than the non adjacent bands.

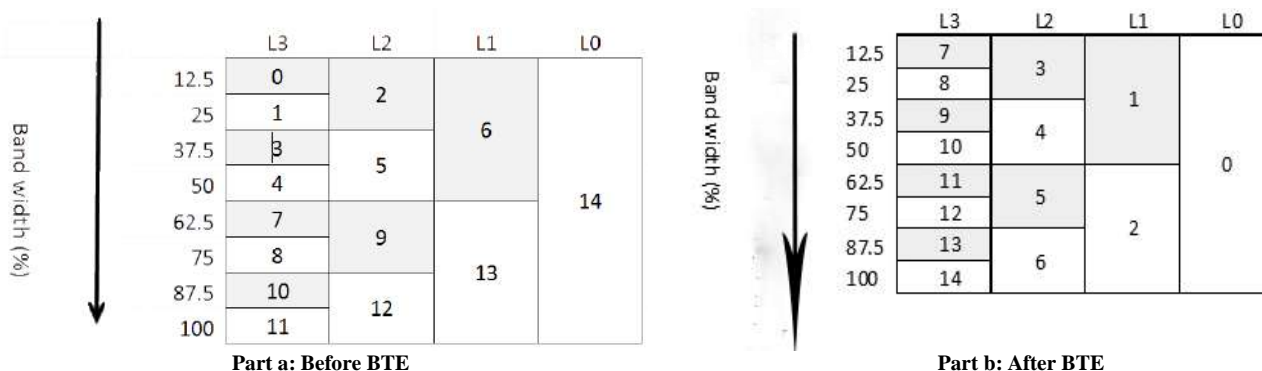


Figure 2: BTE bands are aligned such as to make adjacent wavelet bands are closer in distance than non adjacent bands.

Figure 2-a illustrates how bands are sorted according to Matlab wavelet packets function. Figure 2-b indicates how bands are encoded in BTE. Bands are rearranged for calculating the BTE of the frame. The tree is Encoded into a single number that held information of tree structure {leaves} and weight according to figure 2-b.

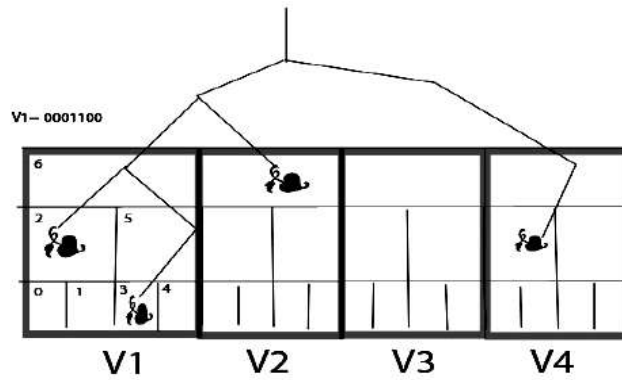


Figure 3: BTE for certain wavelet packets Best tree structure

The indicated tree structure in figure 3 will be encoded into features vector of 3 elements as shown in table 2.

TABLE 2

BEST TREE 4 POINT ENCODING EVALUATION.

Element	Binary Value	Decimal value	Frequency Band
V1	0001100	12	0 - 25 %
V2	1000000	64	25% - 50%
V3	0000000	0	50%-75%
V4	0000100	4	75%- 100%

Features BTE vector ζ for this example of speech frame will be $\zeta = \begin{bmatrix} 12 \\ 64 \\ 0 \\ 4 \end{bmatrix}$

5 EXPERIMENT FRAMEWORK

The framework we developed to train and test GMM HMM models uses HTK to do feature extraction and build the baseline models which are used to align the training data. Microsoft C# (C sharp) is used for building the needed programs and algorithms for building initial models of HTK. HTK tools for training and decoding is a collection of command-line options such as HREst and HVite. Each make a special function, which is explained in detail in HTK book [9]. The phonetic context tree of the HTK baseline models is utilized in proposed system. Training and testing in the proposed system based on Weighted Finite State. Htk tools evaluate the Viterbi path based on likelihood.

6 AUTOMATIC ANNOTATION EXPERIMENTAL PROCEDURE

A. Database Preparation

- Corpus of 300 Arabic sentences of 30 persons (males) sampling rate of 32 kb/s is used. All samples are manually annotated.
- The Database is split into two groups of 150 sentences each. Group A is for training and Group B is for testing.

B. Features Extraction

- All samples are processed to generate MFCC -39 points feature. HTK is used in this step.
- All samples are processed to generate BTE -4 points feature. Matlab is used in this step.

C. Marshaling

All feature files are normalized for being processed in HTK. This process is called marshaling. The data from different sources are rearranged in a way that to be understood by HTK tools. BTE feature vectors files are marshaled into HTK format. HTK allows for user defined features type. This will give HTK tools the ability to be used to process data from other sources not just HTK tools.

D. *Model Design*

- a. 5 nodes LR HMM model is created to model a single phone.
- b. Survey for the most frequently used Gaussian Mixture count for MFCC is used to set the number of Gaussian Mixtures of MFCC model.
- c. For BTE; Gaussian mixture count is an experiment parameter. It will be tuned for the best success rate.
- d. Dictionary and Grammar files will be created for HTK phone recognition problem.
{Illustrate the Grammar file and the dictionary by a graph and a table that clarify the Grammar network and the dictionary}

E. *Training the Models.*

- a. Using HTK and the training samples for MFCC, MFCC models will be trained.
- b. Using HTK and the training samples for BTE, BTE models will be trained.

F. *Testing the models.*

- a. Using HTK and the testing samples for MFCC, MFCC models will be tested.
- b. Using HTK and the testing samples for BTE, BTE models will be tested.

G. *Results*

- a. Results are tabulated for MFCC based recognizer.
- b. Results are tabulated for BTE based recognizer.

Table 3 illustrates the results obtained from both systems. As of the results BTE-4 indicates very comparable results to the well known MFCC features. BTE is still in the development phase. This makes it very promising. BTE is 4 components compared to 39 components of MFCC, makes it a very promising features.

TABLE 3
BTE-4 VERSES MFCC-39 RECOGNITION RESULTS

Feature Type	% Correct	N	I	D	S
BTE-4	57.2%	49	0	17	4
MFCC-39	65.1%	49	0	15	2

N: the total number of labels in the reference transcriptions

I: Number of Insertions errors in the results string.

D: Number of deletion errors in results string.

S: Number of substitution errors in results string.

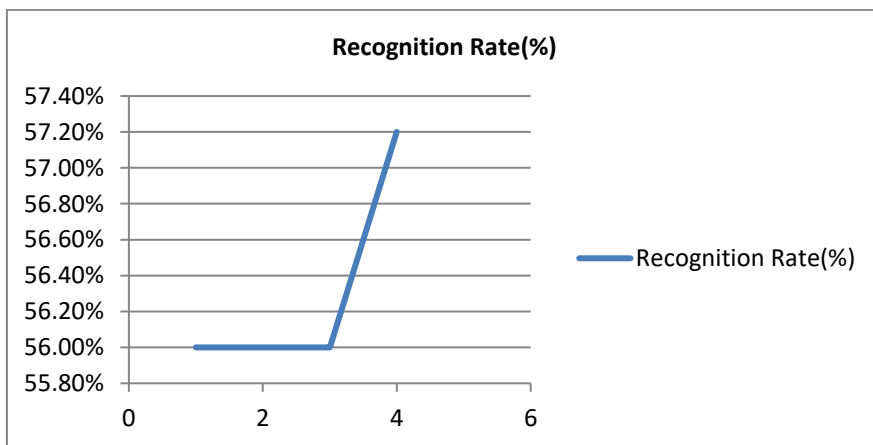


Figure4: Recognition Rate versus Max Number of Mixtures

The number of GM is a factor in the success rate. This number is altered as an experiment parameter. Figure 4 gives the results of changing this value on the success rate.

7 CONCLUSIONS

The results tabulated in table 1 indicate that BTE with 4 components is very promising. BTE is newly developed features that rely on the spectral information. It is a 4 components that is used to encode the whole spectral information of the signal. It gives very close results to the well known feature MFCC with 39 components. This makes it very promising to enhance to give much more efficient results than MFCC.

REFERENCES

- [1] Amr M. Gody, "Wavelet Packets Best Tree 4-Points Encoded (BTE) Features", the 8th Conference on Language Engineering. 2008, Cairo, Egypt.
- [2] Iosif Mporas, Todor Ganchev, Nikos Fakotakis, "Phonetic segmentation using multiple speech features", International Journal of Speech Technology, Springer Netherlands, Volume 11, Number 2 / June 2008, PP. 73-85
- [3] Kris Demuyne, Tom Laureys, "A Comparison of Different Approaches to Automatic Speech Segmentation", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 2448/2006, ISBN 978-3-540-44129-8, PP. 385-406
- [4] Z. M. šarić, S. R. Turajlić, "A new approach to speech segmentation based on the maximum likelihood", Journal of Circuits, Systems, and Signal Processing, Birkhäuser Boston, Volume 14, Number 5 / September 1995, PP. 615-632
- [5] Chin-Teng, Der-Jenq, Rui-Cheng, Gin-Der, "Noisy Speech Segmentation/Enhancement with Multiband Analysis and Neural Fuzzy Networks", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 2275/2002, ISBN 978-3-540-43150-3, PP. 81-94.
- [6] Yanxiang Chen, Qiong Wang, "A Speaker Based Unsupervised Speech Segmentation Algorithm Used in Conversational Speech", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 4798/2007, ISBN 978-3-540-76718-3, PP. 396-402.
- [8] Amr M. Gody, "Voiced/Unvoiced and Silent Classification Using HMM Classifier based on Wavelet Packets BTE features", the 8th Conference on Language Engineering. 2008, Cairo, Egypt.
- [9] Steve Young, Mark Gales, Xunying Andrew Liu, Phil Woodland, et al., 2006. The HTK Book, Version 3.41, Cambridge University Engineering Department, <http://www.htk.eng.cam.ac.uk>.
- [10] Gales, M.J.F., Young, S.J., 2007. *The application of hidden Markov models in speech recognition*. Foundations and Trends in Signal Processing (3), 195–304.
- [11] Young, S., Odell, J.J., Woodland, P.C., 1994. *Tree-based state tying for high accuracy acoustic modeling*. In: Proc. 1994 ARPA Human Language Technology Workshop, pp. 304–312.

Implementation of Establishing Global Ontology by Matching and Merging

Susan F. Ellakwa ^{*1}, Passent El-Kafrawy^{**2}, Mohamed Amin^{**}, El-Sayed El-Azhary^{*3}

^{*}Central Lab for Agricultural Expert Systems (CLAES), ARC, Giza, Egypt

¹fisalsusan@yahoo.com

³sayed@claes.sci.eg

^{**}Mathematics and CS Department, Faculty of Science, Menoufia University, Egypt

²passentmk@gmail.com

Abstract— Ontology is used for communication between people and organizations by providing a common terminology over a domain. This work presents implementation of the system of establishing global ontology by matching and merging. Establishing ontology from scratch is hard and expensive. This work establishes ontology by matching and merging existing ontologies. Ontologies can be matched and merged to produce a single integrated ontology. Integrated ontology has consistent and coherent information rather than using multiple ontologies, which may be heterogeneous and inconsistent. Heterogeneity between different ontologies in the same domain is the primary obstacle for interoperability between systems. Heterogeneity leads to the absence of a standard terminology for any given domain that may cause problems when an agent, service, or application uses information from two different ontologies. Integrating ontologies is a very important process to enable applications, agents and services to communicate and understand each other.

Keywords: Artificial Intelligence, Knowledge Representation, Ontology, Matching, Merging.

1 INTRODUCTION

The term ontology refers to a wide range of formal representations, including taxonomies, hierarchical terminology vocabularies or detailed logical theories describing a domain [1]. One commonly used definition is based on the original use of the term in philosophy, where ontology is a systematic account of Existence. For artificial intelligence (AI) systems, what “exists” is that what can be represented [2]. "An Ontology is a formal, explicit specification of a shared conceptualization [3]. *Conceptualization* refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. *Explicit* means that the type of concepts used, and the constraints on their use, are explicitly defined. *Formal* refers to the fact that the ontology should be machine-readable. *Shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.

There are several reasons for developing ontology. First, sharing common understanding of the structure of information among people or software agents. Second, enabling the reuse of knowledge. Third, making domain assumptions explicit. Fourth, separating domain knowledge from the operational knowledge. Fifth, analyzing domain knowledge. Sixth, increasing interoperability among various domain of knowledge. Seventh, enhancing scalability of new knowledge into the existing domain. Finally, searching and reasoning a specific knowledge in domain knowledge.

This paper presents implementation of the system of establishing global ontology by matching and merging [13]. Global ontology allows users to avoid querying the local ontologies one by

one, and to obtain a result from them just by querying a global ontology. Global ontology has standard and shared terminology. It is consistent and coherent. It has no redundancy.

There are a large variety of languages for expressing ontologies. Fortunately, most of these languages share the same kinds of entities, often with different names but comparable interpretations. Source ontologies in the proposed system have been expressed in XML language. Ontology language in the proposed system deal with the following kinds of entities: Concepts, properties, and values according to CommonKADS Methodology [4].

In this system, we introduce an ontology matching and merging problem and propose an implementation for Multi-Matching and Merging Algorithm (MMMA) [13], which uses a multi search algorithm to find the correspondences between entities in the input ontologies and to merge these ontologies. An important feature of this technique is that it benefits from existing individual match methods and combines their results to provide enhanced ontology matching.

This system proposes a new technique in matching; it performs three iterations, each iteration manipulates one type of entities. The first iteration manipulates the concepts, while the second iteration handles the properties, and the third iteration handles the values. In each iteration, the system uses hybrid matchers which are combined in a sequential composition. This multilevel decomposition reduces redundancy alignments and speeds up the system's final alignments. The system uses different kinds of matchers to cover different kinds of alignments to reduce redundant entities of resulted merged ontology. Using variety of matchers solve the string and language matching problem. This system extracts entities in two ontologies which have same string or same meaning. The system uses thresholds to reduce useless alignments and involves user to confirm alignments. This system can merge the ontologies in hierarchy structure.

This paper consists of five sections; first section is introduction, second section shows definition for matching and merging, third section introduces related work, fourth section presents the implementation of the proposed system in [13] and its graphical interface and fifth section is conclusion and future work.

2 ONTOLOGY MATCHING AND MERGING

Matching is the process of finding relationships or correspondences between entities of different ontologies. Alignment is a set of correspondences between two or more (in case of multiple matching) ontologies. The alignment is the output of the matching.

The matching process can be seen as a function f which, from a pair of ontologies to match o and o' , an input alignment A , a set of parameters p and a set of oracles and resources r , returns an alignment A' between these ontologies:

$$A'=f(o, o', A, p, r)$$

The proposed system uses the matching techniques; string-based technique [5] (String equality method, Substring method and Prefix/suffix method) and language-based technique [5] (tokenization method, Stopword elimination method and WordNet [6] method) as blocks on which a matching solution is built. Each of these methods is called a matcher. Each matcher gives its similarity. Once the similarity between ontology entities is available, the alignment remains to be computed.

Merging is a first natural use of ontology matching, it consists of obtaining a new ontology o'' from two matched ontologies o and o' so that the matched entities in o and o' are related by the alignment. Merging can be presented as the following operator:

$$\text{Merge}(o, o', A) = o''$$

When the ontologies are expressed in the same language, merging often involves putting the ontologies together and generating bridge or articulation axioms. Merging does not usually require a total alignment: those entities which have no corresponding entity in the other ontology will remain unchanged in the merged ontology. Ontology merging is especially used when it is necessary to carry out reasoning involving several ontologies. It is also used when editing ontologies in order to create ontologies tailored for a particular application.

3 RELATED WORK

Several tools exist for ontology establishment, ranging from fully manual to fully automated. Many of the semi-automated ontology merging and matching tools are listed in this section. PROMPT[7] begins with the linguistic-similarity matches for the initial comparison, but generates a list of suggestions for the user based on linguistic and structural knowledge and then points the user to possible effects of these changes.

OntoMorph [8] provides a powerful rule language for specifying mappings, and facilitates ontology merging and the rapid generation of knowledge-base translators. It combines two powerful mechanisms for knowledge-base transformations such as syntactic rewriting and semantic rewriting. Syntactic rewriting is done through pattern-directed rewrite rules for sentence-level transformation based on pattern matching. Semantic rewriting is done through semantic models and logical inference. A concept hierarchy management for ontology alignment and merging is provided in Hierarchical Concept Alignment system (HICAL) [9], where one concept hierarchy is aligned with another concept in another concept hierarchy. HICAL uses a machine-learning method for aligning multiple concept hierarchies, and exploits the data instances in the overlap between the two taxonomies to infer mappings. It uses hierarchies for categorization and syntactical information, not similarity between words, so that it is capable of categorizing different words under the same concept. Another system that employs machine learning techniques to find ontology mappings is GLUE [10]. If given two ontologies, for each concept in one of the ontologies, GLUE finds the most similar concept in the other one. GLUE works with several similarity measures that are defined with probabilistic definitions. Multiple learning strategies exploit different types of information from instances or taxonomy structures. GLUE can also use common sense knowledge and domain constraints instead of relaxation labeling. It is a well-known constraint optimization technique adapted to work efficiently. Quick Ontology Mapping (QOM) [11] is based on the hypothesis that mapping algorithms can be streamlined so that the loss of quality is marginal, but the improvement of efficiency is tremendous for the ad-hoc mapping of large-size light-weight ontologies.

A generic ontology mapping system, called LILY [12], is based on the extraction of semantic subgraph. LILY exploits both linguistic and structural information in semantic subgraphs to generate initial alignments. After that, a subsequent similarity propagation strategy is applied to produce more alignments if necessary. Finally, LILY uses the classic image threshold selection algorithm to automatically select the threshold, and then extracts final results based on the stable marriage strategy. LILY has different functions for different kinds of tasks: for example, Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies; Large scale Ontology Matching method (LOM) is used for matching tasks with large size ontologies; and Semantic Ontology Matching method (SOM) is used for discovering the semantic relations between ontologies. The two limitations of LILY are that it requests the user to manually set the size of subgraph according to different mapping tasks and the efficiency of semantic subgraph is very low in large-scale ontologies.

4 SYSTEM IMPLEMENTATION FOR ESTABLISHING GLOBAL ONTOLOGY

This section presents implementation of the proposed system in [13] to get global ontology from existing ontologies. It proposed also a case study. The system has been implemented by using ASP.NET C#. This system has been applied on ontologies represented in XML language.

The proposed system interface contains the following topics: Home page, Match ontologies, Merge ontologies, Stopword, Preview ontology, and Edit ontology.

A. Home Page

This is the main window of the system, see figure 1.



Figure 1: Home Window

B. Match Ontologies

This topic presents implementation of matching process with its interface. The interface consists of three iterations; first iteration is to get concept alignments, second iteration is to get property alignments and third iteration is to get value alignments. Each iteration has five matchers: exact matcher, substring matcher, prefix matcher, suffix matcher and WordNet matcher.

Figure 2 presents first window in 'Match Ontologies', the user can browse to determine the source ontologies. The built in value of similarity threshold is 0.5, but the user can determine it if he wants (see Figure 3).



Figure 2: Match Window

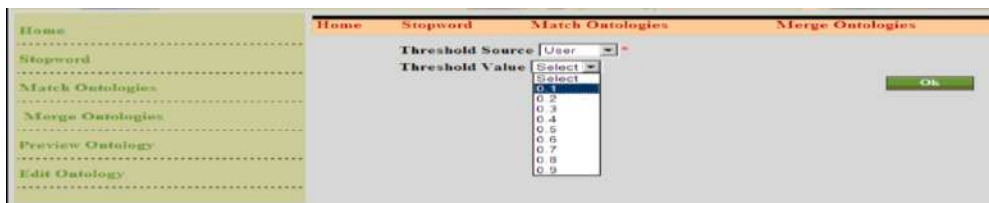


Figure 3: Threshold value is 0.1

o Concept alignment interface

Concept alignment interface shows implementation of the first iteration for five matchers on concepts of source ontologies. Figure 4 shows interface of substring matcher for concepts. Figure 5 shows interface of prefix matcher for concepts. Figure 6 shows interface of suffix matcher for concepts. Figure 7 shows interface of WordNet matcher for concepts. Figure 8 shows interface of concepts alignments.

Threshold Source *

Threshold Value

Current concepts matcher is : Substring matcher

Suggestion For Matchable Concepts	Similarity
<input checked="" type="checkbox"/> flowers Aligns To flower	0.92
<input checked="" type="checkbox"/> fruits Aligns To fruit	0.91
<input checked="" type="checkbox"/> pests Aligns To pest	0.89
<input checked="" type="checkbox"/> roots Aligns To root	0.89
<input checked="" type="checkbox"/> viruses Aligns To virus	0.83
<input checked="" type="checkbox"/> the-plantpart Aligns To plantpart	0.82
<input type="checkbox"/> pesticide Aligns To pest	0.62

Figure 4: Output of substring matching for concepts

Figure 5: Output of prefix matcher for concepts

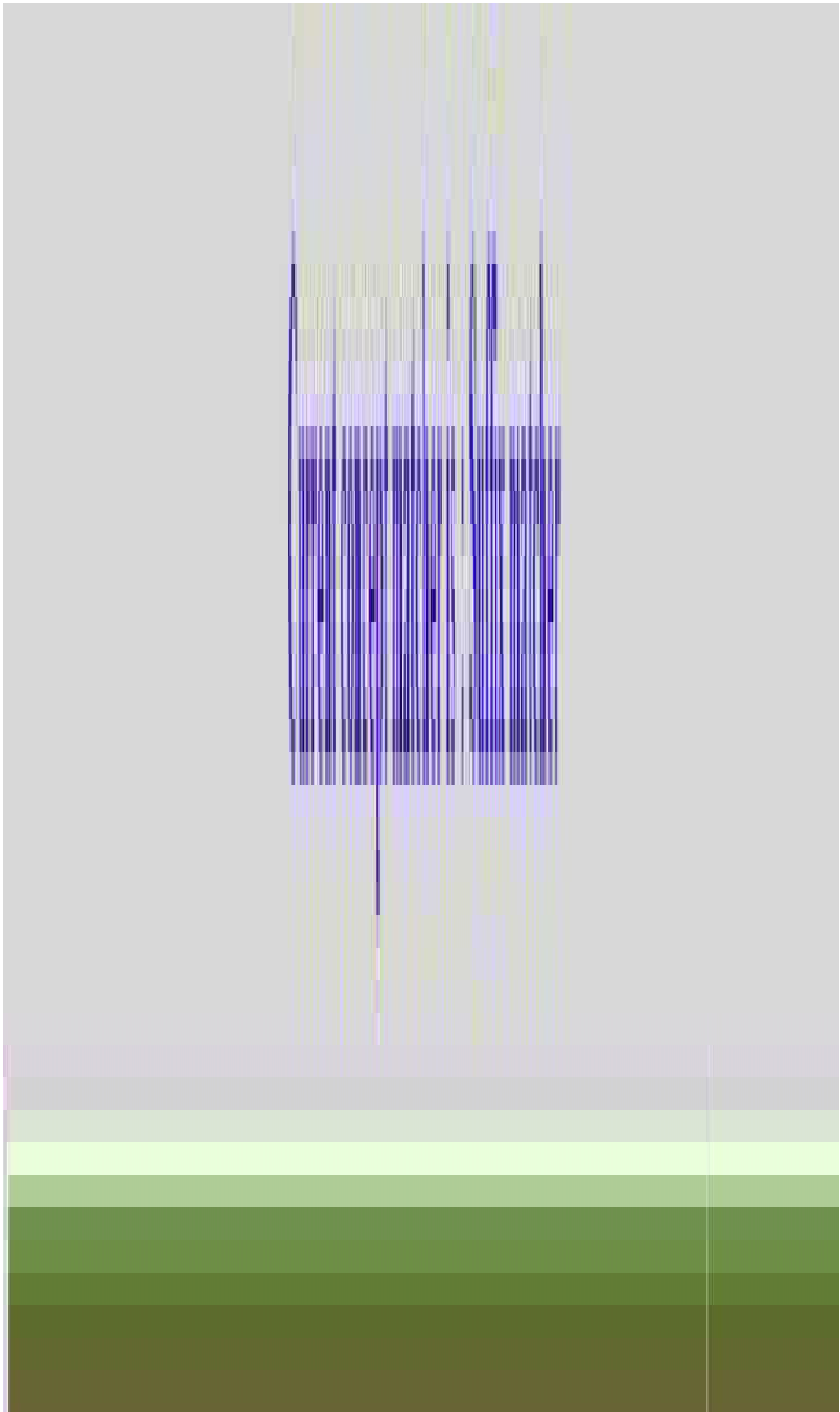


Figure 6: Output of suffix matcher for concepts

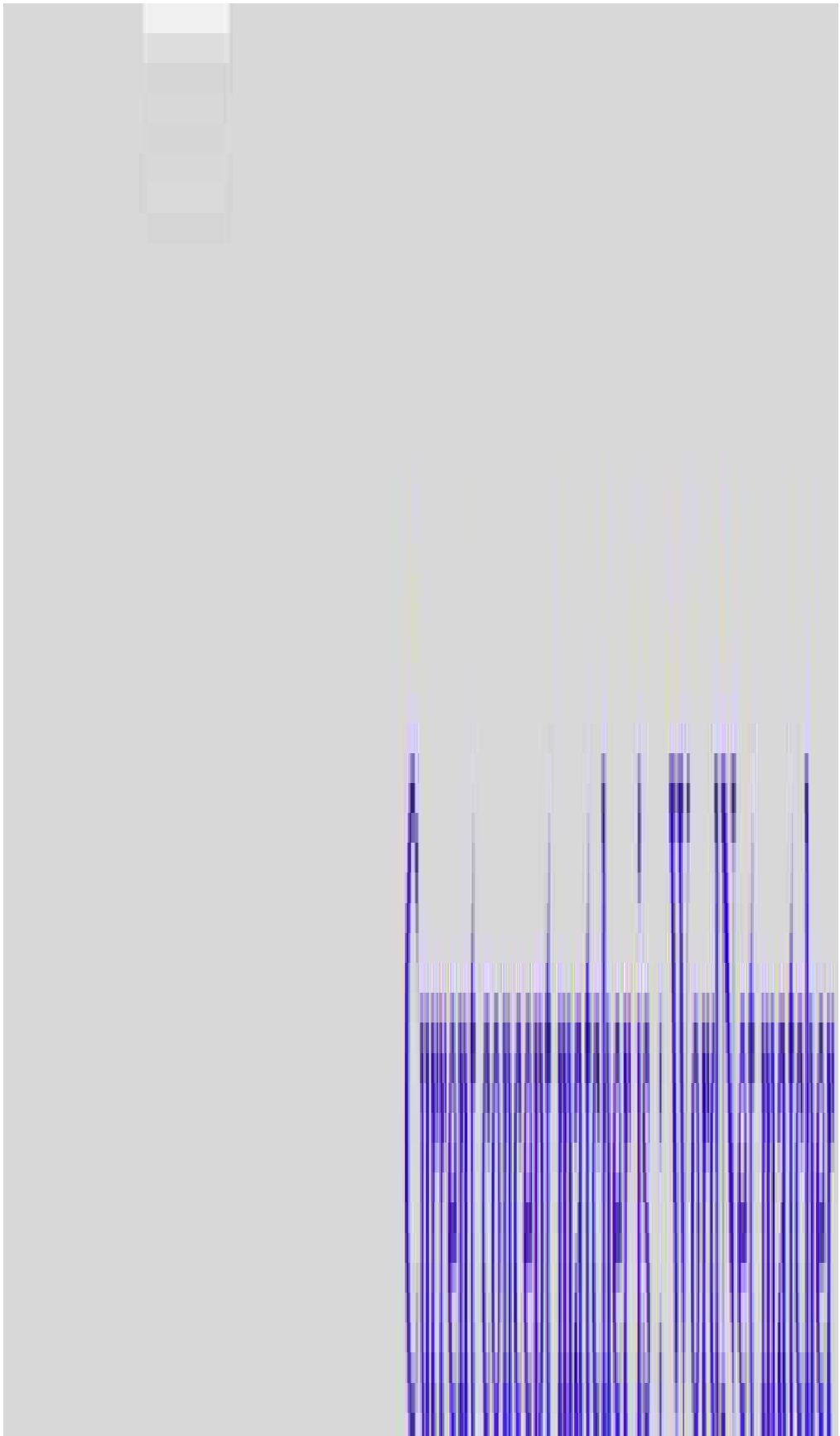


Figure 7: Output of WordNet matcher for concepts

Alignments For Matched Concepts	
base	Aligns To the-stem
water-system	Aligns To irrigation-system
ontology2	Aligns To ontology1
fungi	Aligns To fungus
plant-name	Aligns To plants
leaves	Aligns To leaf
flowers	Aligns To flower
fruits	Aligns To fruit
pests	Aligns To pest
roots	Aligns To root
viruses	Aligns To virus
the-plantpart	Aligns To plantpart
pesticide	Aligns To pesticide
insect	Aligns To insect
Match properties	

Figure 8: Concept Alignment

○ *Property alignment interface*

Property alignment interface shows implementation of the five matchers on properties of source ontologies. Figure 9 shows interface of substring matcher for properties. Figure 10 shows interface of prefix matcher for properties. Figure 11 shows interface of suffix matcher for properties. Figure 12 shows interface of WordNet matcher for properties. Figure 13 shows interface of properties alignment

Current properties matcher is : Substring matcher		
Suggestions For Matchable Properties		
<input checked="" type="checkbox"/>	shapes Aligns To flower_shape	0.90000
<input checked="" type="checkbox"/>	leaf_shape Aligns To fruit_shape	0.80000
<input checked="" type="checkbox"/>	leaf_shape Aligns To leaf_shape	0.80000
<input checked="" type="checkbox"/>	name Aligns To pesticide_the_name	0.66667
Next Match:		

Figure 9: Output of substring matcher for properties

current properties matcher is : Prefix matcher		
Suggestions For Matchable Properties		
<input checked="" type="checkbox"/>	quantities Aligns To irrigation-system_quantity	0.7
<input checked="" type="checkbox"/>	color Aligns To flower_colour	0.66667
<input checked="" type="checkbox"/>	color Aligns To fruit_colour	0.66667
<input checked="" type="checkbox"/>	color Aligns To root_colour	0.66667
<input checked="" type="checkbox"/>	color Aligns To leaf_colour	0.66667
Next Match:		

Figure 10: Output of prefix matcher for properties

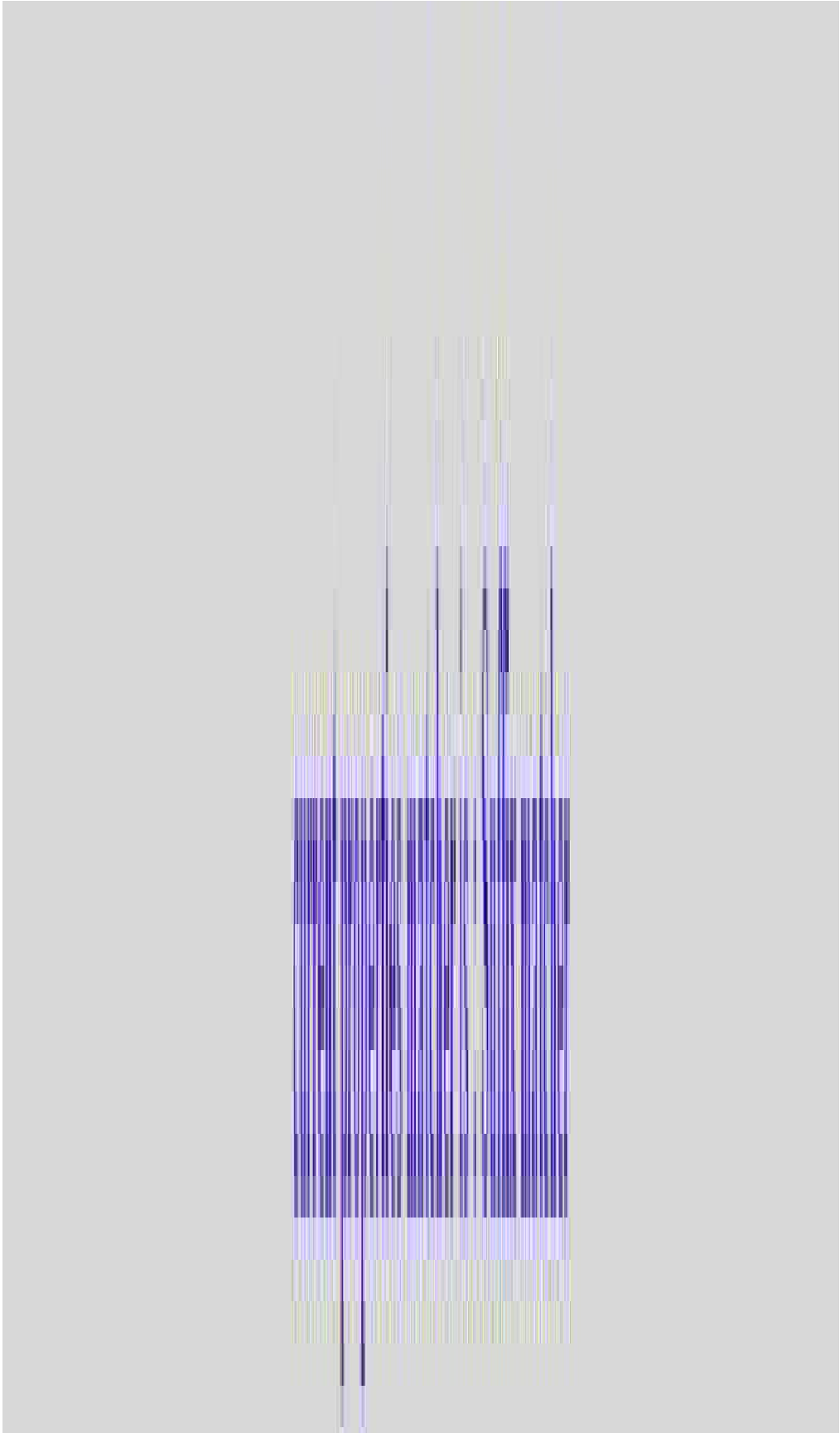


Figure 11: Output of suffix matcher for properties



Figure 12: Output of WordNet matcher for properties

Alignments For Matched Properties
pesticide . name Aligns To pesticide .the-name
insect . sort Aligns To insect .kind
flowers . age Aligns To flower .age
flowers . shapes Aligns To flower .shape
flowers . color Aligns To flower .colour
fruits . age Aligns To fruit .age
fruits . shapes Aligns To fruit .shape
fruits . color Aligns To fruit .colour
roots . shapes Aligns To root .shapes
roots . age Aligns To root .age
roots . color Aligns To root .colour
viruses . sort Aligns To virus .kind
fungi . sort Aligns To fungus .kind
leaves . age Aligns To leaf .age
leaves . shapes Aligns To leaf .shape
leaves . color Aligns To leaf .colour
water-system . quantities Aligns To irrigation-system .quantity
water-system . water-schedule Aligns To irrigation-system .irrigation-schedule
water-system . sort Aligns To irrigation-system .kind
Match values

Figure 13: Property Alignment

○ *Value alignment interface*

Value alignment interface shows implementation of the five matchers on values of source ontologies. Value alignment interface shows implementation of the five matchers on properties of source ontologies. Figure 14 shows interface of substring matcher of values. Figure 15 shows interface of prefix matcher for values. Figure 16 shows interface of suffix matcher for values. Figure 17 shows interface of WordNet matcher of values. Figure 18 shows interface of Final Alignment.

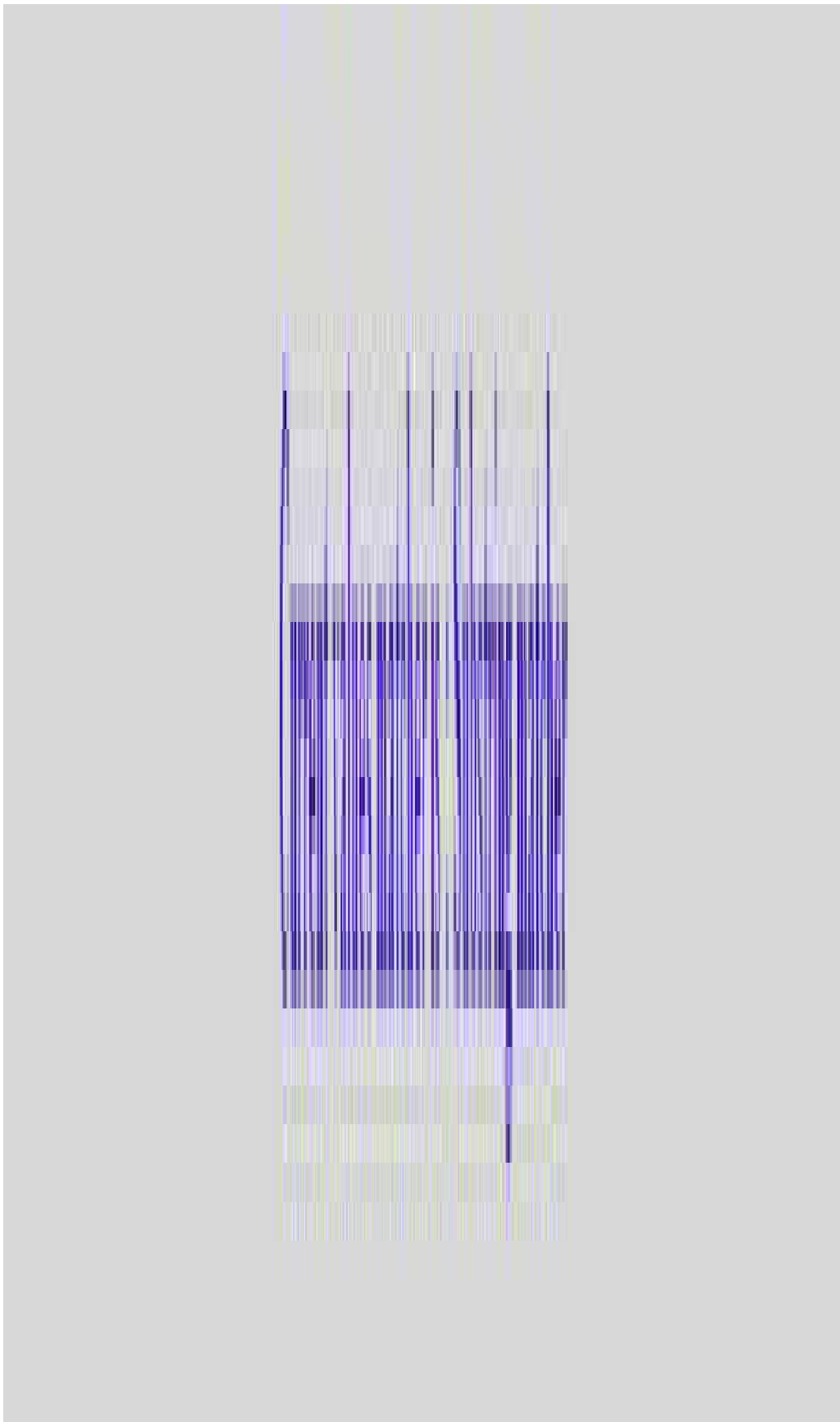


Figure 14: Output of substring matcher for values



Figure 15: Output of prefix matcher for values

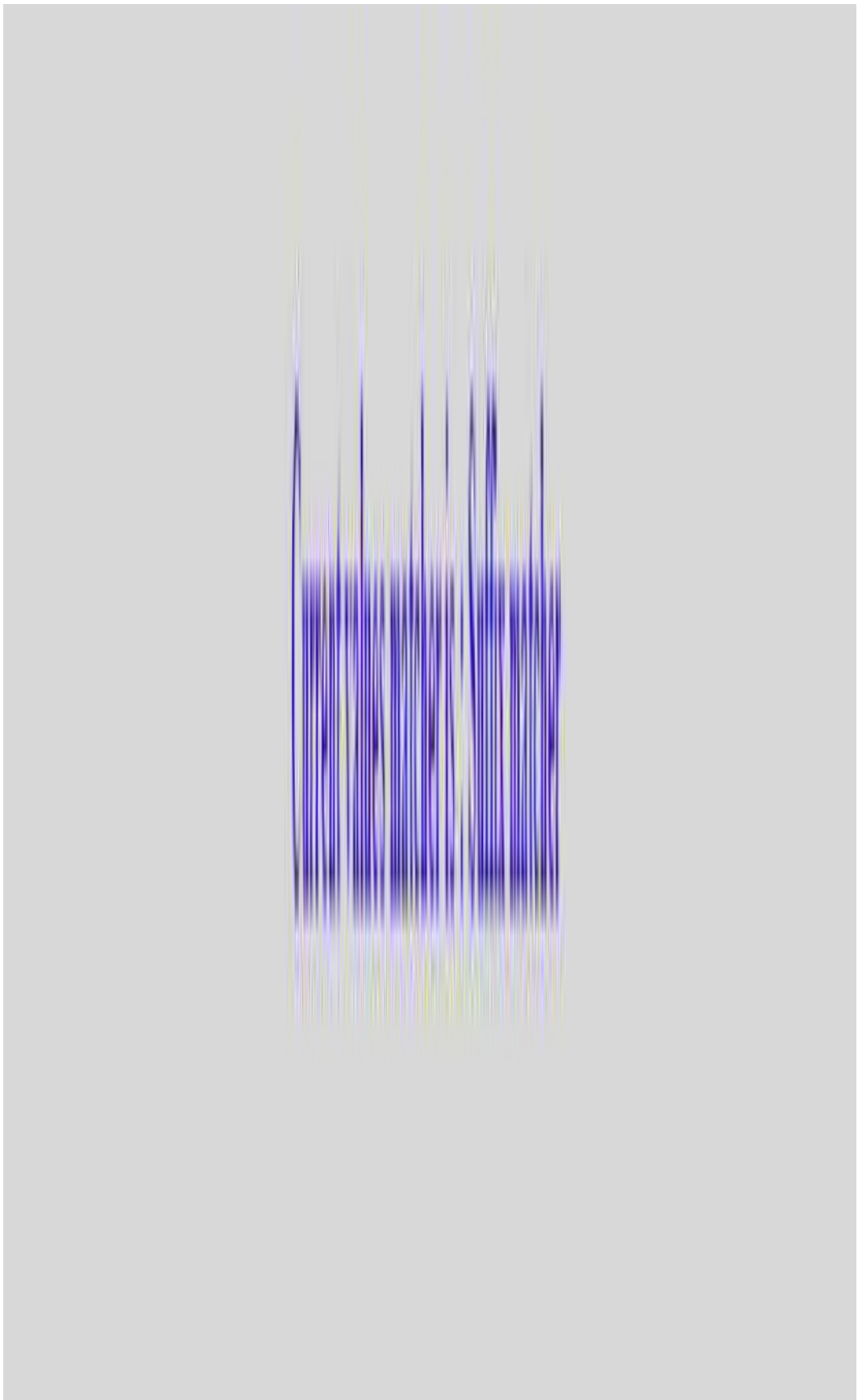


Figure 16: Output of suffix matcher for values

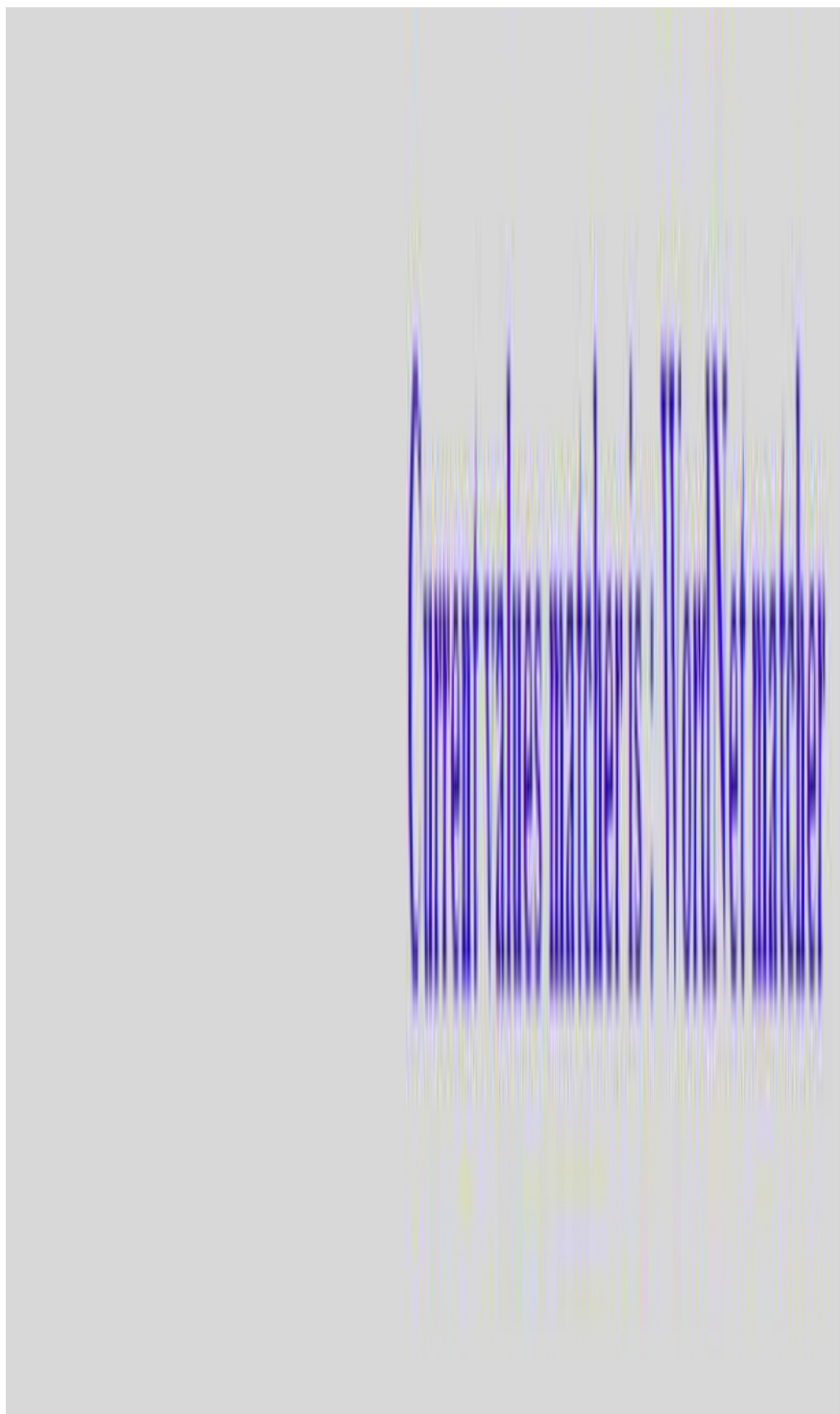


Figure 17: Output of WordNet matcher for values

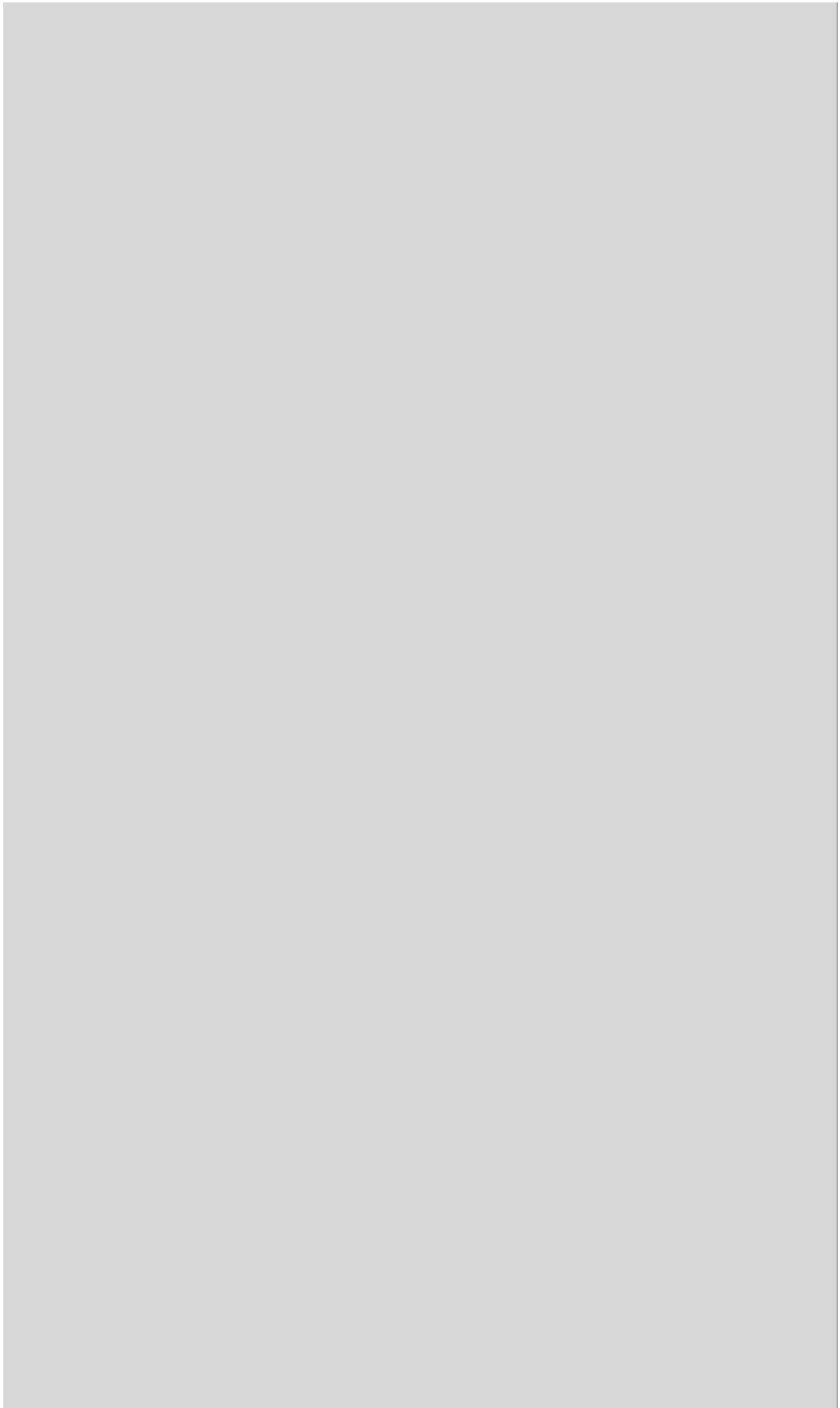


Figure 18: Final Alignment

C. Merge Ontologies

This section presents implementation of merging process with its interface. The input of this process is the source ontologies besides the output of the matching process: concepts alignment, properties alignment and values alignment. The output is the merged ontology. The window in Figure19 is to name merged ontology. Figure 20 shows the merged ontology and ontologies information.

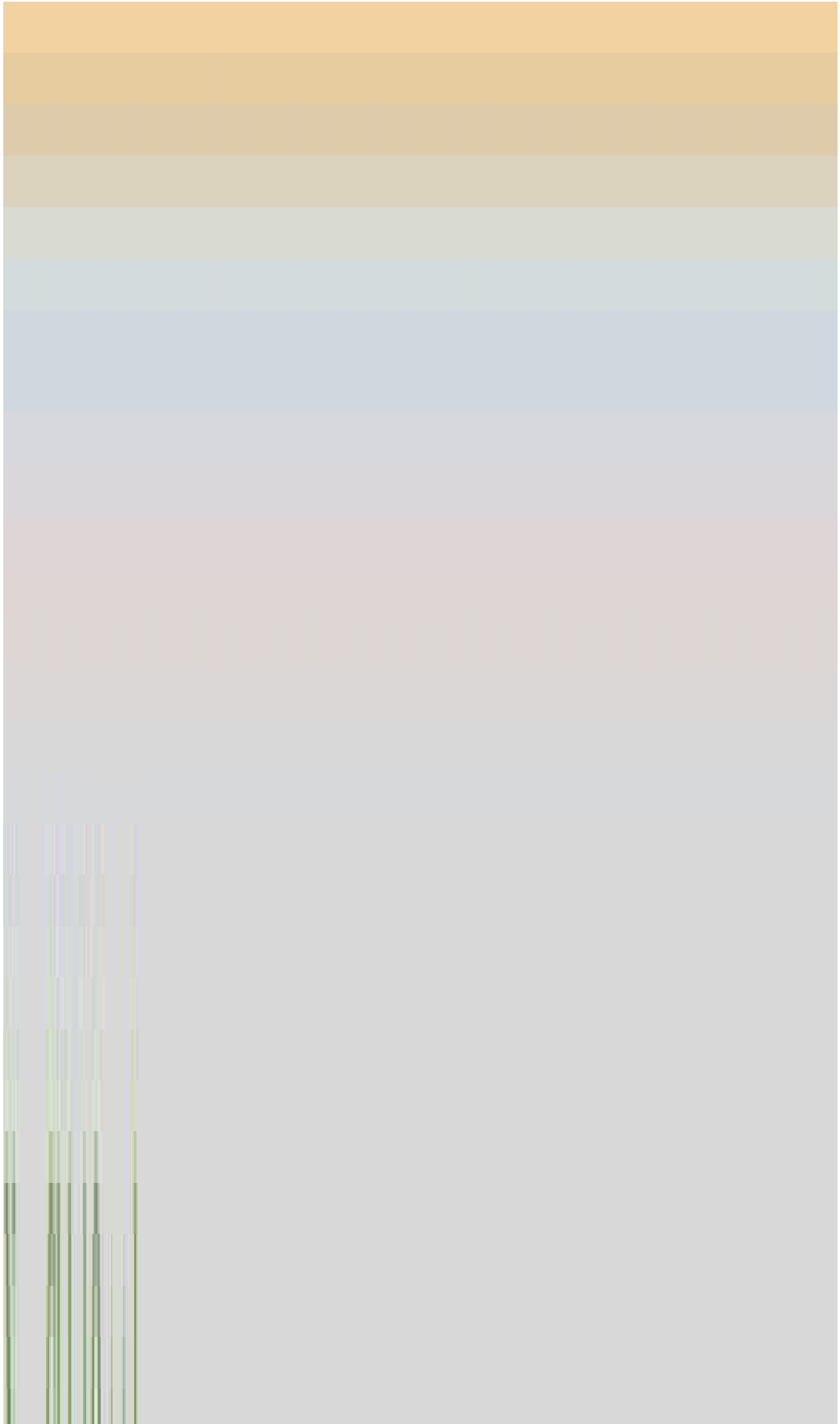


Figure 19: Merge source ontologies

Name: Global Ontology
 Select Global Ontology: MergedOntology10102011.xml Stop Delete

Ontologies Information			
	First Source Ontology (ontology_2_XML.xml)	Second Source Ontology (ontology1_XML.xml)	Global Ontology (MergedOntology10102011.xml)
Concept Number	14	18	18
Property Number	22	21	24
Value Number	15	29	29
Global Ontology File			

Figure 20: Merged ontology and ontologies information

D. Preview Ontology

This topic presents source ontologies in Figure 21 and Figure 22 and merged ontology in hierarchal structure (see Figure 23)

Select Ontology file:

Concepts	Properties	Values
<ul style="list-style-type: none"> <input type="checkbox"/> ontology1 <ul style="list-style-type: none"> <input type="checkbox"/> pest <ul style="list-style-type: none"> <input type="checkbox"/> insect <input type="checkbox"/> virus <input type="checkbox"/> fungus <input type="checkbox"/> pesticide <input type="checkbox"/> plants <ul style="list-style-type: none"> <input type="checkbox"/> plantpart <ul style="list-style-type: none"> <input type="checkbox"/> the-stem <input type="checkbox"/> root <input type="checkbox"/> flower <input type="checkbox"/> fruit <input type="checkbox"/> leaf <input type="checkbox"/> irrigation-system <input type="checkbox"/> climate 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> quantity <input type="checkbox"/> irrigation-schedule <input type="checkbox"/> kind 	<ul style="list-style-type: none"> <input type="checkbox"/> value <input type="checkbox"/> time-irrigation

Figure 21: Preview first source ontology

Select Ontology file:

Concepts	Properties	Values
<ul style="list-style-type: none"> <input type="checkbox"/> ontology1 <ul style="list-style-type: none"> <input type="checkbox"/> pest <ul style="list-style-type: none"> <input type="checkbox"/> insect <input type="checkbox"/> virus <input type="checkbox"/> fungus <input type="checkbox"/> pesticide <input type="checkbox"/> plants <ul style="list-style-type: none"> <input type="checkbox"/> plantpart <ul style="list-style-type: none"> <input type="checkbox"/> the-stem <input type="checkbox"/> root <input type="checkbox"/> flower <input type="checkbox"/> fruit <input type="checkbox"/> leaf <input type="checkbox"/> irrigation-system <input type="checkbox"/> climate 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> quantity <input type="checkbox"/> irrigation-schedule <input type="checkbox"/> kind 	<ul style="list-style-type: none"> <input type="checkbox"/> value <input type="checkbox"/> time-irrigation

Figure 22: Preview second source ontology

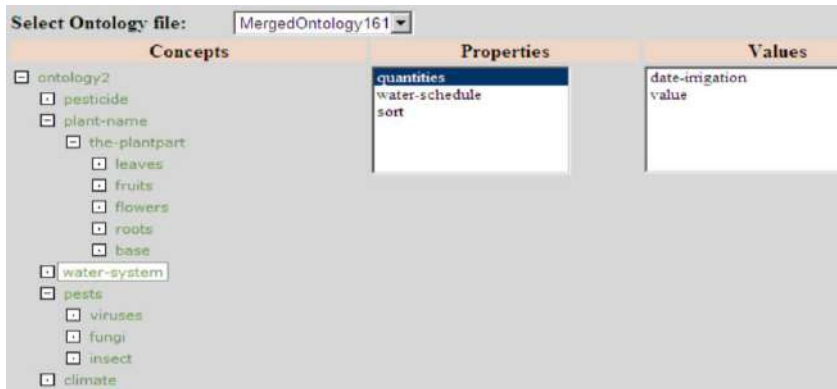


Figure 23: Preview Merged Ontology

E. *Edit Ontology*

This topic presents interface of manipulating the entities of source ontologies and merged ontologies. The user can add new entities, delete or update existing entities. Figure 24 shows interface of manipulating entities of ontologies

F. *Stopword Elimination*

This topic presents interface of entering discard words, they are considered as non meaningful (empty) words for WordNet matcher. Figure 25 shows interface of Stopword Elimination.

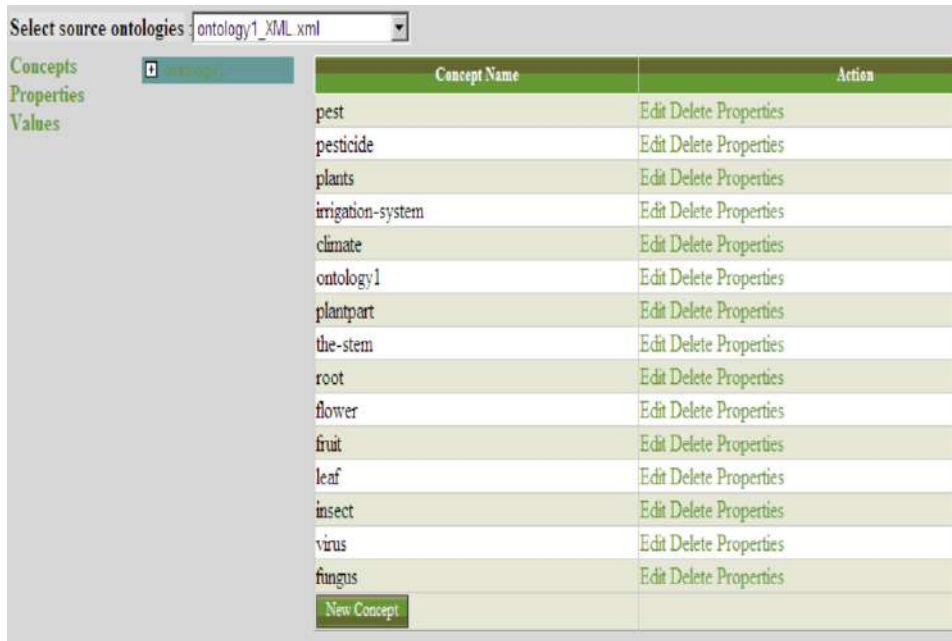


Figure 24: Edit Ontology Window

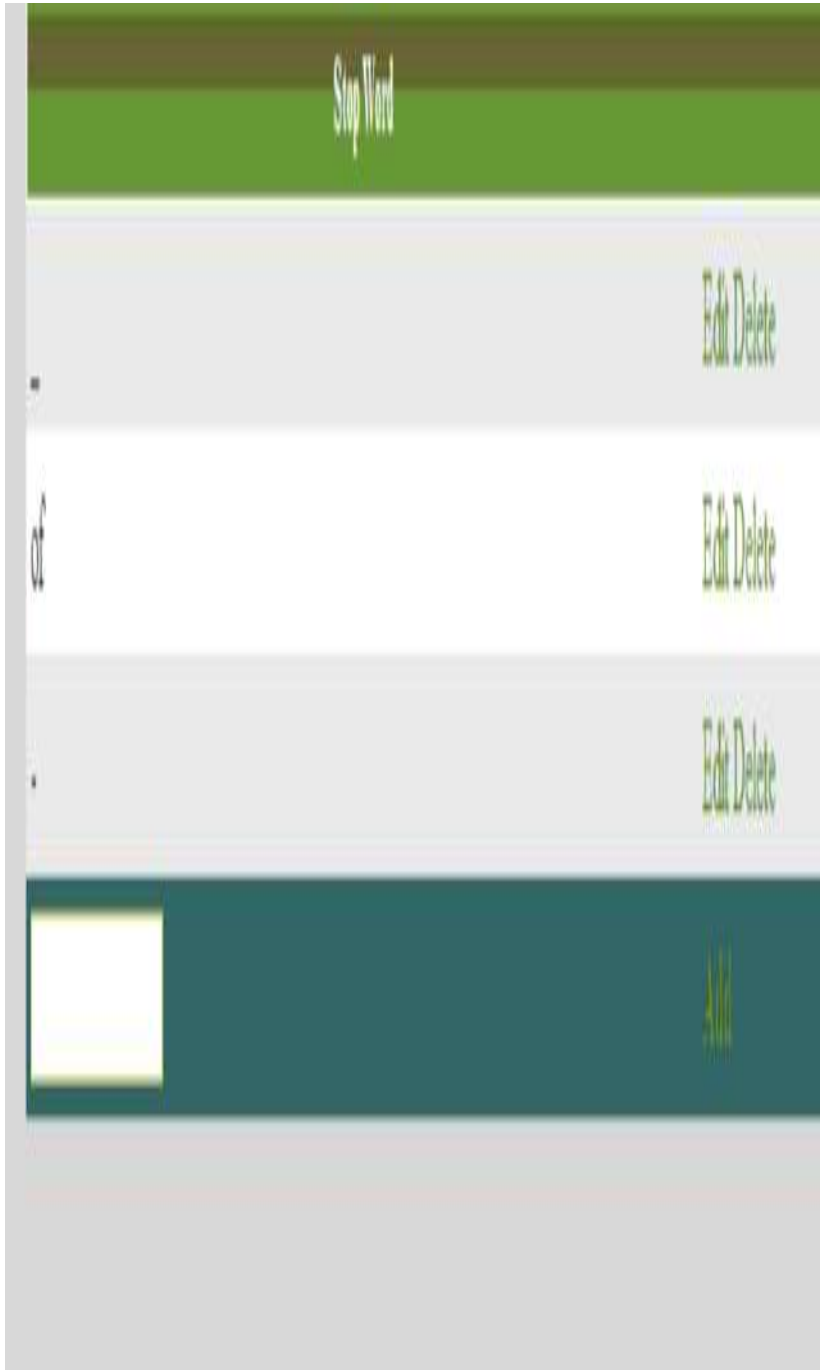


Figure 25: Stopword elimination window

Building global ontology from scratch is hard, cost and time-consuming; this paper presents implementation of establishing global ontology from different ontologies in the same domain by matching and merging. It presents a case study of the proposed system. It demonstrates the different steps for building the global ontology. The system have graphical user interface to allow browsing to get ontologies to be matched and merged. It allows user to confirm alignments, edit and preview source ontologies and merged ontology, it gives information about source ontologies and merged ontology.

References

- [1] Noy, N., Klein, M.: “Ontology Evolution: Not the Same as Schema Evolution”, *Knowledge and Information Systems*, 6 (4), pp. 428-440 (2004), also available as SMI technical report SMI-2002-0926.
- [2] Gruber, T.R. “A Translation Approach to Portable Ontology Specifications”, *Knowledge Acquisition*, 5 (2), pp.199-220 (1993).
- [3] Borst P, Akkermans H, Top J.: “Engineering ontologies”, *International Journal of Human-Computer Studies* 46:pp.365–406,1997.
- [4] Bon J. Wielinga, *Expertise Model Definition Document*, University of Amsterdam, 1994.
- [5] Euzenat, J., Shvaiko, P. “Ontology matching”, *Springer Verlag*, Heidelberg (DE), 333 p., 2007.
- [6] Pedersen, T., Patwardhan, S., Patwardhan, S. “WordNet::Similarity – Measuring the Relatedness of Concepts”. In *Proc. of 19th National Conf.on AI*, San Jose, 2004.
- [7] N. Noy and M. Musen,: “PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment”, *Proc.of17th National Conference on Artificial Intelligence,(AAAI)* , pp. 450–455, Austin, Texas, 2000.
- [8] H. Chalupsky. “Ontomorph: A Translation System for Symbolic Knowledge”, *Principles of Knowledge Representation and Reasoning*, 2000.
- [9] R. Ichise, H. Takeda, and S. Honiden. “Rule Induction for Concept Hierarchy Alignment”, *Proceedings of the Workshop on Ontology Learning at the 17th International Joint Conference on Artificial Intelligence (IJCAI)* 2001.
- [10] An-Hai Doan, J. Madhavan, Pedro Domingos, and Alon Halevy. “Learning to map ontologies on the semantic web”, In *Proceedings of the International World Wide Web Conference (WWW)*, pages 662–673, 2003.
- [11] Marc Ehrig and Steffen Staab. “QOM: Quick ontology mapping”, In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 683–697, 2004.
- [12] Peng Wang and Baowen Xu. “Lily: Ontology alignment results for oaei 2009”.In Shvaiko. 2009.
- [13] Susan F. Ellakwah , Passent El-Kafrawy, Mohamed Amin, El Sayed ElAzhary. “Establishing Global Ontology by Matching and Merging”. *To appear in SPIT-ASP-Springer proceeding 2011*, Amsterdam, Netherlands, 2011.

SAFAR Platform and its Morphological Layer

Younes Souteh and Karim Bouzoubaa

*Mohammadia School of Engineers, Mohammed Vth University - Adgal,
Rabat, Morocco.*

y_souteh@yahoo.fr
karim.bouzoubaa@emi.ac.ma

Abstract— The development of natural language processing applications leads in general to the development of many aspects of the language starting from the morphological level, the syntactic one, the semantic and even the pragmatic ones. Such development requires the use of multiple tools of each aspect of the language. In the present work, we show how SAFAR as an integrated Arabic language processing platform can be used to handle several aspects of the language.

1 INTRODUCTION

One of the main issues to consider when developing any natural language processing (NLP) application is the choice of the most appropriate tool. For the particular case of the Arabic language, many interesting developments tools already exist: morphological analyzers to define the structure of words [1, 2, 3, 4], stemmers to reduce a word down (or some derivative) to its root or its radical [5, 6, 7], or also parsers that determine the syntax of phrases [8, 9]. In most cases, the development of Arabic Natural Language Processing (ANLP) applications requires the use of several tools at once, each dealing with a certain level of language (morphology, syntax, semantics and pragmatics). For example, to develop an automatic translator, one approach necessitates the use of an analyzer as well as a morphological parser. Generally these tools are developed by different teams with different programming languages. Also, and very often, the output of one tool is not directly exploitable by another tool. For example, ALKhalil analyzer [1] outputs the result in an HTML page that is not directly usable by other applications. Therefore, the ANLP application developer must very often face problems of integration of different technologies, a more difficult maintenance of the system, a larger number of codes and a tedious search of the most appropriate tools. Thus, to avoid such difficulties, it would be interesting to have a single integrated environment allowing researchers to develop different aspects of the language and that offers:

- Basic ANLP modules including morphological, syntactic and semantic tools for each one of these aspects
- Free resources (dictionaries, corpora, lexical database, etc.)
- Resources and modules for comparison and evaluation
- Technical basic services (tokenizer, vowels removal)

This article describes how the SAFAR platform (Software Architecture For Arabic language pRocessing) addresses the above needs. SAFAR is a Java and an open source platform dedicated to the ANLP development, providing the foundation for integrated process solutions for Arabic language. It is in our view an effective way for standardization, optimization efforts, collaboration and accelerating developments in the area.

The rest of this article is as follows. Section 2 describes existing and most known platforms in the ANLP field. Section 3 is dedicated to the description of the SAFAR platform. Section 4 is a focus on the stemming service and section 5 is dedicated to the morphology analyzer service. Finally, the last section concludes the work and discusses some future horizons.

2 EXISTING PLATFORMS

Before starting on the design and development of SAFAR, we reviewed the existing works that could address the need to have a platform that processes generally any natural language and can be adapted to the case of Arabic.

GATE (General Architecture for Text Engineering) [10] is an infrastructure of development and deployment components for natural language processing. Developed since 1995 at the University of Sheffield, it is widely used on text mining and information extraction tasks. GATE provides an architecture, a framework in Java (including many modules) and an integrated development environment. However, the GATE component are too abstract and does not propose a specification in terms of API and components output compliant with ANLP needs, which does not promote the integration of existing tools. In addition,

it does not propose a layered architecture compliant with ANLP levels (morphology, syntax, and semantics). In addition to its use, Gate requires a considerable cost to learn a language called JAPE used to model rules.

NooJ [11] is a linguistic development environment for building, testing and maintaining natural languages formalized descriptions (as electronic dictionaries and grammars), and developing language processing applications. But it adopts a single formalism (analysis model), based on automata, and is based on pipeline architecture to form complex processing. As GATE, it does not propose a specification of ANLP components.

UIMA (Unstructured Information Management Architecture) [12] is a software architecture for the development and deployment of tools for analyzing unstructured information. Its purpose is to describe the steps for processing a text document, image or video to automatically extract structured information. However, UIMA does not describe how this information must be extracted, or how to use it. The aim of this very general environment makes its architecture very abstract. Consequently, no analysis module for language automatic processing is used immediately. The implementation of treatment for a particular task remains the responsibility of the designer, who must find analysis components developed by himself or by third parties, which remain at present relatively rare and very specific.

"Two Tools" is a family of platforms that combines several complementary ANLP tools for specific treatments of the Arabic language. An example of these platforms is MADA-Tokan [13] that incorporates morphological analysis of a word regardless of context and morphological disambiguation to choose the solution depending on the context. The platform AMIRA [14] is also part of this family. It includes tools for segmentation, annotation of parts of speech and syntactic analysis.

Thus, we can summarize our literature review that all the platforms listed above do not provide integrated and coherent specification of ANLP modules. Therefore, the direct use of these modules is limited and calls for further development by the programmer. SAFAR aims to overcome these limitations for the various needs of the ANLP community.

3 SAFAR

SAFAR [15.16] is a platform dedicated for ANLP. It is open source, portable, modular, extensible, flexible and offers an integrated development environment (IDE). Figure 1 gives an overview of its architecture.

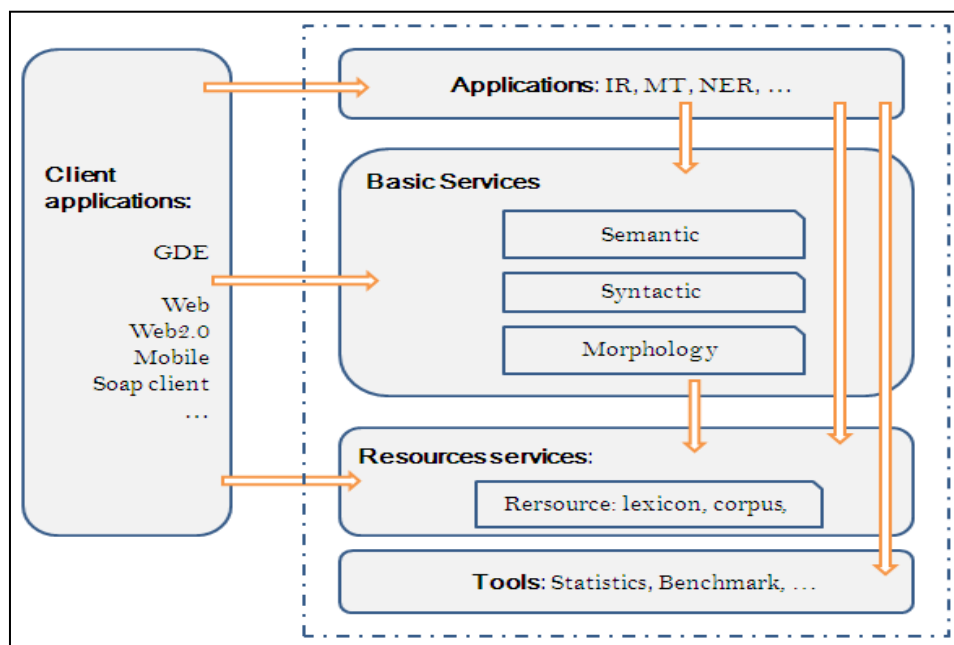


Figure 1: Architecture of SAFAR platform

Each layer is developed as a set of reusable Java APIs that provide services directly usable by the other layers in accordance with the use relationship (modeled with arrows in the figure). These layers are:

- Tools: Includes a set of technical services
- Resources Services: Provides services for consulting language resources such as lexicons and corpora
- NLP Services: Contains the three regular layers for processing a language (morphology, syntax and semantics)
- Application: Contains high-level applications that use the layers listed above.
- Client applications: Contains the client applications that can directly use the services of one or more layers.
-

In general, our philosophy is not to develop ourselves all the SAFAR layers and modules, but to integrate existing ones consistently. Consequently our approach consists in providing the specification in terms of APIs for each module of our architecture and also provide (if any) implementations of these APIs with applications that have proved to be efficient. For example, the Buckwalter morphological analyzer [17] is very popular within the ANLP community and it would be interesting to continue using it as part of a new platform. Thus, several principles follow from this approach:

- Reuse: this feature is used to introduce a level of interoperability between the different modules of the platform. This allows to create composites modules from existing ones
- Portability: All modules are developed in Java
- Open: All modules are specified through an API that complies with the linguistic rules of the Arabic language. The use of APIs allows to standardize the development of new modules and to integrate existing applications through adapters with the proviso of respecting the interface of the API. Thus, it is possible to integrate existing applications or to propose new implementations with respect to a module's API.
- Open source: it could be used and evaluated by the community
- Flexibility of exploitation: it is possible to use the platform in various ways either in local mode in the form of Java archive, or in remote mode through web services or through a graphical development kit that includes a set of plug-ins that facilitate the development with a drag & drop system

The first layer we have implemented in SAFAR is the morphological layer. Respecting the principles of SAFAR, the development of this layer is structured with Java interfaces (APIs) and with some known implementations. This layer includes two families of modules: the generators that produce forms of words using morphological attributes and analyzers that identify the components of a word which in turn are organized into two modules: a stemmer and an morphology analyzer . So far we have achieved the last two modules.

4 STEMMER

Stemming is a process to remove all prefixes and suffixes of a word to produce a stem or root [5]. Its importance appears in the creation of indexes that speed up the information retrieval algorithms by bringing together all words that share the same stem (or the same root).

Stemming algorithms are classified into two types [20]:

- Stem-based Algorithms (light stemmer) [6] that remove affixes (prefix and suffix). Several implementations exist: Aljlal & Frieder's [20], Darwish's Al-Stem [21], Chen & Gey's TREC 2002 Stemmer [22], and Larkey et al.'s U Mass Stemmer [23]
- Root-based Algorithms (aggressive) that retrieve the root of a word [24]. Several implementations exist such as Khoja stemmer [25] and Darwish stemmer [26]

To integrate existing implementations of stemmers, we selected the most commonly used: Khoja [25] and Al-Stem [23]. Khoja Stemmer is developed in Java and uses lists of patterns and roots to retrieve the root of a given word. Although it is well referenced in the ANLP world, it has limitations in terms of usability:

- It does not offer an API that facilitates its integration and the only possibility is to run it in GUI mode.
- The result of an analysis is not directly usable.

Al-stem was built by kareem Darwish and modified by Leah Larkey at the University of Massachusetts [26]. As the majority of light Stemmers, it begins with a normalization to remove the diacritics followed by a removal of affixes. Al-stem is developed in Perl language which limits its integration with Java applications in addition to its unformatted output that does not help its exploitation.

To develop the Stemmer, we began, firstly, by specifying an API (figure 2) with two methods to analyze the word and the text (respectively, `stem` and `stemText`) and a class model to represent the result of analysis which is composed of the Morpheme and its type (root, stem).

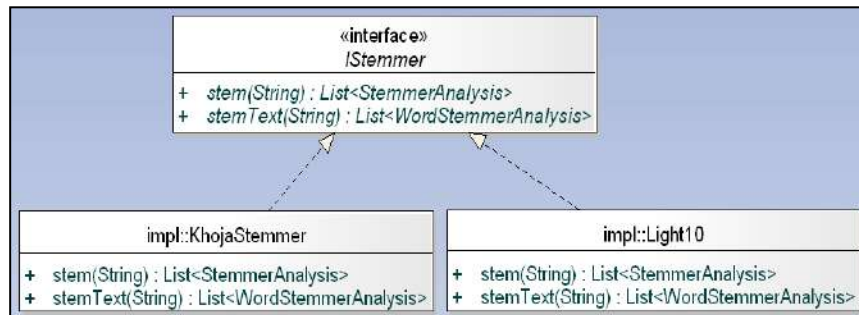


Figure 2: Stemmer API

For implementing the two stemmers (Khoja Stemmer and AlStem), we formatted the output of The Khoja Stemmer so that it is supported by the SAFAR API and rewrote Al-Stem in Java language. Figure 3 shows an example of using the SAFAR Stemmer with the Khoja implementation.

```

package test.safar.morphology.stemmer.impl;

import java.util.List;
import junit.framework.TestCase;
import safar.morphology.interfaces.IStemmer;
import safar.morphology.stemmer.impl.KhojaStemmer;
import safar.morphology.stemmer.model.StemmerAnalysis;
import safar.morphology.stemmer.model.WordStemmerAnalysis;

public class KhojaStemmerTest extends TestCase {

    public void testStemText() {
        String text = "هل ينتظر الغداني غايات وتوسلات و استغاثات لكي يتعطف على" +
            " "الأسر البائسة ليظلمتهم على شياهم المفقود" +
            " "nالديه منذ سبع سنوات لا يعلمون عنهم شيئاً";
        IStemmer stemmer = new KhojaStemmer();
        List<WordStemmerAnalysis> result = stemmer.stemText(text);
        for(WordStemmerAnalysis wordStemmerAnalysis:result){
            for(StemmerAnalysis stemmerAnalysis:wordStemmerAnalysis.getListStemmerAnalysis()){
                addToXMLOutput(stemmerAnalysis);
            }
        }
    }

    private void addToXMLOutput(StemmerAnalysis stemmerAnalysis) {}
}
  
```

Figure 3: Example of SAFAR Stemmer API based on Khoja implementation

The program in Figure 3 imports the class implementation `KhojaStemmer`, calls the constructor and the `stemText()` method to analyze the text and goes through the different solutions to output them in an XML format. The stemmer specification with a corresponding API provides more flexibility in terms of operations. Figure 4 shows some of the text analyzed by the program.

```

- <word value="هل">
- <stemAnalysis>
  <solution number="1" morpheme="هل" type="STOPWORD" />
</stemAnalysis>
</word>
- <word value="ينظر">
- <stemAnalysis>
  <solution number="1" morpheme="نظر" type="ROOT" />
</stemAnalysis>
</word>
- <word value="الغذائي">
- <stemAnalysis>
  <solution number="1" morpheme="غذ" type="ROOT" />
</stemAnalysis>
</word>
- <word value="التحليلات">
- <stemAnalysis>
  <solution number="1" morpheme="حول" type="ROOT" />
</stemAnalysis>
</word>
- <word value="وتوسلات">
- <stemAnalysis>
  <solution number="1" morpheme="وسل" type="ROOT" />
</stemAnalysis>
</word>

```

Figure 4: output stemmer in XML format (Khoja implementation)

5 MORPHOLOGICAL ANALYZER

The morphological analyzer is an essential component of many NLP applications. On the one hand, it can support various applications for the end user (spell checker, online dictionaries, etc.), and on the other hand it can constitute a basis for the syntactic and semantic layers. There are several implementations of morphological analyzers. In this section we detail those we selected and we consider to be among the most commonly used: Buckwalter [17] and Alkhalil analysers [1].

Buckwalter morphological analyzer (Aramorph) was developed in two versions, Perl and Java. Although widely used, it has from the SAFAR point of view the following limitations [18]:

- It does not have an API that facilitates its integration with other systems, the only possibility is to run it from the command line and use the results manually.
- Its output is not directly used: the analysis result is generated in text format with a specific structure encoded in a transliterated format [19].

Alkhalil (Alkhalil Morpho Sys) [1] was developed in Java. It is freely available as open source. It was chosen by ALECSO¹ (Arab League Educational, Cultural and Scientific Organization) as their reference analyzer. However, it presents the same limitations as those of Aramorph with an analysis output in HTML format. This limits its integration with other systems and necessitates the development of new utilities to exploit it.

For the morphological analyzer, we adopted an approach that respects the SAFAR principles. We began by specifying the API of the analyzer and then we integrated the Buckwalter and Alkhalil implementations.

A. API Specification

For this part, we need to define:

- Firstly, the API functions that returns either morphological properties such as the stem, suffix, root, etc. without a full analysis of the word or complete analysis of a word or set of words. Each analysis offers several solutions each consisting of several segments (prefix, radical, suffix), a type, part of speech (POS) and morphological attributes.
- Secondly, the parts of speech and morphological attributes. We used the Alecso proposal to identify these elements that we consider to be more compatible with the rules of the Arabic language.

We therefore propose an API for the Java interface with several functions (Figure 5).

¹ www.alecso.org

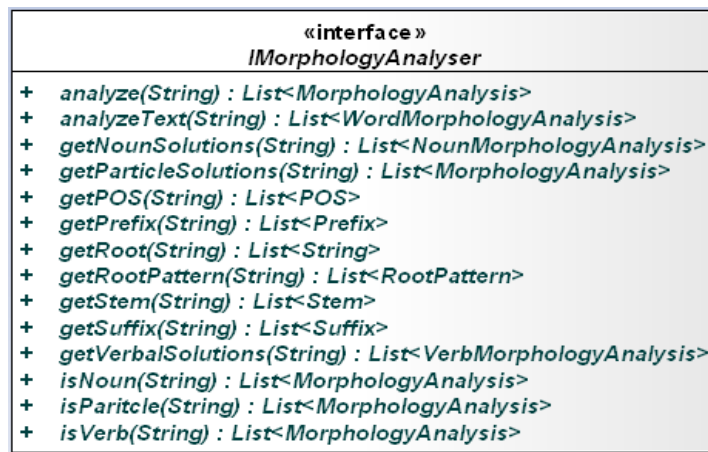


Figure 5: Morphology analyzer API

The following class diagram (Figure 6), presents the objects results of the complete analysis of a word and a text (respectively WordAnalysis and MorphologyAnalysis):

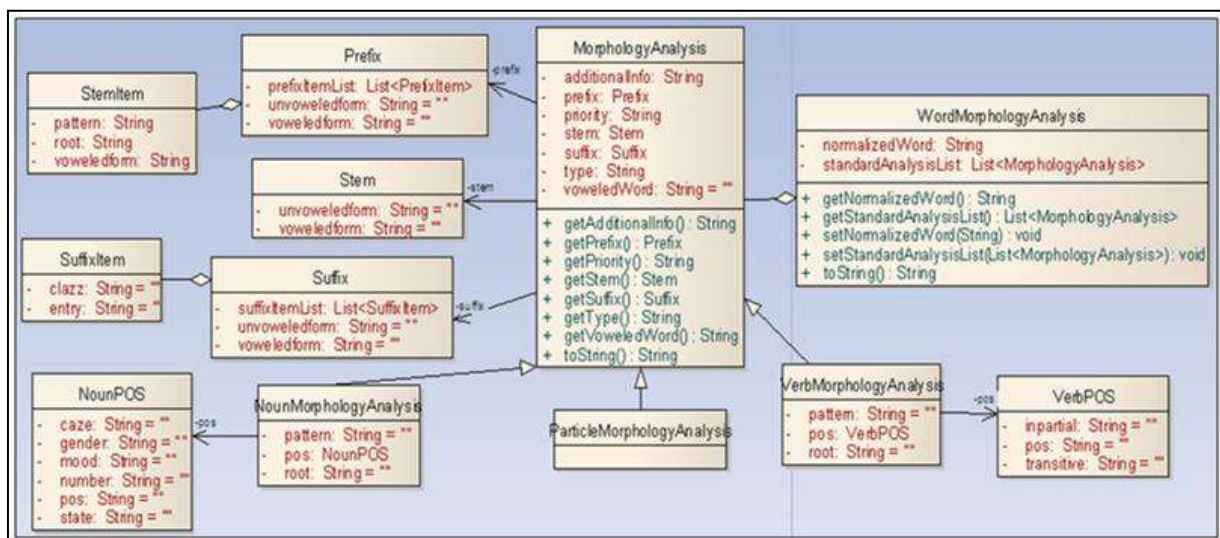


Figure 6: class model of morphology analyzer of SAFAR platform

From the above diagram, each analysis result is composed of three morphological units:

- Prefix is a type of affixes that is placed at the beginning of a stem such as: ك, س, ف, ب
- Radical (stem) which is the smallest token that represents the main part of a word
- suffix which is a type of affixes that is placed at the end of a stem such as ... نا, ها, هم

In addition, the analysis result contains morphological attributes by word type (noun, verb or particle). For example, for type "noun" in addition to the prefix, suffix and the radical, it defines the morphological attributes: pattern, POS and root. For the types "noun" and "verb", we associate respectively NounPOS and VerbPOS. We have also added accessors (getters) for the different properties of the result (for example, morphologyAnalysis.getPrefix() which returns the prefix)

B. API implementation

To implement the interface IMorphologyAnalyser of our analyzer, we must define all these methods. For our two analyzers (Buckwalter and Alkhalil), we have developed "adapters" to transform their results of analysis to the format defined for the model of the SAFAR morphological analyzer. The use of interfaces, we guarantee more flexibility in

choosing the most suitable implementation. Figure 7 shows an example of using the API of the analyzer: to use a given implementation simply import the corresponding class without impact on the rest of the code.

```

package test.safar.morphology.analyzer.impl;

import java.util.List;
import junit.framework.TestCase;
import safar.morphology.analyzer.impl.AlkhalilMorphologyAnalyser;
import safar.morphology.analyzer.model.MorphologyAnalysis;
import safar.morphology.interfaces.IMorphologyAnalyser;

public class AlkhalilMorphologyAnalyserTest extends TestCase {
    public void testAnalyzeWord() {
        IMorphologyAnalyser analyser = new AlkhalilMorphologyAnalyser();
        String word = "بأصواتهم";
        List<MorphologyAnalysis> list = analyser.analyze(word);
        for (MorphologyAnalysis morphologyAnalysis : list) {
            addToXMLOutput(morphologyAnalysis);
        }
    }

    private void addToXMLOutput(MorphologyAnalysis morphologyAnalysis) {
        // TODO Auto-generated method stub
    }
}

```

Figure 7: Example of SAFAR morphology analyzer API based on Alkhalil implementation

The program in Figure 7 imports the class implementation `AlkhalilMorphologyAnalyser`, calls the constructor and the `analyze()` method to analyze the word before it goes through the various solutions and produces an output in a XML format (Figure 8). This example shows the advantage of using interfaces to provide more flexibility and use the results of analysis to multiple outputs.

```

<word value="بأصواتهم">
  <morphologyAnalysis>
    <solution number="1" type="اسم جامد" pos="مذكر مجرور في"
    "صوت" root="أصوات" stem="أفَعَان" pattern="حالة الاضافة
    prefix="[بم حرف الجر]" suffix="[مير الفاعلين]"/>
    <solution number="2" type="مصدر مرة" pos="مؤنث مجرور في"
    "ءمو" root="أصوات" stem="فَعَلَات" pattern="حالة الاضافة
    prefix="[بم حرف الجر]" suffix="[مير ضمير]"
    </morphologyAnalysis>
  </word>

```

Figure 8: output morphology analyzer in XML format (Alkhalil implementation)

6 CONCLUSION

This article presents the most known NLP platforms such as Gate, NOOJ, UIMA and two-Tools and puts the focus on their limitations as compared to the needs of the ANLP community. We have described the characteristics of our SAFAR platform. Thus, the dimensions of openness and standardization make it a solid foundation to develop and integrate different ANLP solutions and services. Up to now, we realized the software structure of SAFAR and the development of the morphological level (morphological analysis and stemmer). We intend in future works to complete the language resources and services layers, the morphological generator, the other basic layers (syntax and semantics) and the prototype for some applications (such as a search engine, a question answering system, etc.).

REFERENCES

- [1] http://www.alecso.org.tn/index.php?option=com_content&task=view&id=1302&Itemid=998&lang=ar
- [2] Imad Al-Sughayer and Ibrahim Al-Kharashi. "Arabic morphological Analysis Techniques: a comprehensive Survey". Computer and Electronics Research Institute, King Abdul Aziz City for Science and Technology. *Journal of the American Society for information Science and Technology* Februry 1, 2004.
- [3] Kais Dukes and Nizar Habash. Morphological Annotation of Quranic Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC), Malta, 2010*.
- [4] Imad Al-Sughaiyer and Ibrahim Al-Kharashi.. Arabic Morphological Analysis Techniques: A Comprehensive Survey. *Journal of the American Society for Information Science and Technology, 2004*.

- [5] Shereen Khoia. (no date). "APT: Arabic Part-of-speech Tagger". [online]. Available from: <http://archimedes.fas.harvard.edu/mdh/arabic/NAACL.pdf> [Accessed 17/07/03].
- [6] Larkey, L. S., Ballesteros. Improving stemming for Arabic information retrieval: *light stemming and co-occurrence analysis Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, 2002.*
- [7] Darwish, K., Doermann, D., Jones, R., Oard, D., and Rautiainen, M. TREC-10 experiments at Maryland: CLIR and video. In TREC. Gaithersburg: NIST, 2001.
- [8] Richard Socher, Christopher D. Manning. Better Arabic parsing: baselines, evaluations, and analysis, *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics*
- [9] <http://www.cimos.com/index.php?src=list#grammar>
- [10] H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan. GATE: an Architecture for Development of Robust HLT. In Proc. ACL-2002.
- [11] Silberstein, M. 2006 NooJ Manual. Download from "http://www.nooj4nlp.net".
- [12] Ferrucci D., Lally A., « UIMA : an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment », *Natural Language Engineering*, vol. 10, p. 327-348, 2004.
- [13] Nizar Habash and Owen Rambow. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics, June 2005
- [14] Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, pages 149–152, Boston, MA, 2004. DOI
- [15] S. Sidrine, Y. Souteh, K. Bouzoubaa, T. Loukili. SAFAR: vers une plateforme ouverte pour le traitement automatique de la langue Arabe. in the Special Issue on "Advances in Arabic Language Processing" for the International Journal on Information and Communication Technologies (IJICT), Serial Publications, June 2010, 11:2533-2541, 2010.
- [16] L. Abouenour, S. El Hassani, T. Yazidy, K. Bouzoubaa, A. Hamdani. Building an Arabic Morphological Analyzer as part of an Open Arabic NLP Platform. Workshop HLT & NLP within the Arabic world : Arabic Language and Local Languages Processing Status Updates and Prospects, Language Resources and Evaluation Conference LREC'2008, 31st May, Marrakech, Morocco, 2008.
- [17] Buckwalter. ARABIC MORPHOLOGY ANALYSIS. <http://www.qamus.org/morphology.htm>, 2002.
- [18] Mohammed Attia. 'An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks'. The Challenge of Arabic for NLP/MT Conference, October 2006. The British Computer Society, London, 2006.
- [19] <http://www.qamus.org/transliteration.htm>
- [20] M. Aljlal and O. Frieder. 2002. On Arabic Search: Improving the retrieval effectiveness via a light stemming approach. In Proceedings of CIKM'02, VA, USA.
- [21] K. Darwish and D. Oard. 2002. CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English Retrieval. In Proceedings of TREC, Gaithersburg, Maryland, 2002.
- [22] A. Chen and F. Gey. 2002. Building an Arabic stemmer for information retrieval. In Proceedings of TREC, Gaithersburg, Maryland, 2002.
- [23] L. S. Larkey and M. E. Connell. 2001. Arabic information retrieval at UMass. In Proceedings of TREC 2001, Gaithersburg: NIST, 2001
- [24] Jinxi Xu, Alexander Fraser, and Ralph M. Weischedel. "Empirical Studies in Strategies for Arabic Retrieval". In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), pp. 269-274. Tampere, Finland, 2002.
- [25] Shereen Khoja: Arabic stemmer <http://zeus.cs.pacificu.edu/shereen/research.htm#stemming>
- [26] Kareem Darwish. "Al-stem: A light Arabic stemmer" [Online]. Available: <http://www.glue.umd.edu/~kareem/research>, 2002

Correctness, Strength and Similarity Evaluation of Stemming Algorithms for Arabic

Daoud Daoud*¹, Christian Boitet**²

*Princess Sumaya University for Technology

¹d.daoud@psut.edu.jo

**GETALP, LIG, Université Joseph Fourier

²christian.boitet@imag.fr

Abstract--- In this paper, we present a comprehensive evaluation of four Arabic stemmers, based on metrics for correctness, strength and similarity. Two data sets were used in this study. For correctness evaluation, we used a list of 8697 Arabic words grouped into 1606 conceptual classes. For similarity and strength evaluation, we used a list of 72,000 unique Arabic words. Conclusions about correctness, strength and similarity of the four Arabic stemming algorithms are reported.

1 INTRODUCTION

Detecting the surface variations of the same word is one of the main challenges of any type of natural language processing system. Specifically, the effectiveness for information retrieval depends on its ability to map all those variations to the same form.

Stemming is the process of automatically revealing a word's stem. In other words, stemming a word is actually the removal of all the inflectional morphemes from the word's surface-form. Lemmatization goes a step further in identifying the *citation form* of the word, also often called its *lemma*, typically used to access dictionaries. In many languages, the inflected or derived wordforms of a lemma have several stems.

Most researchers in the field of Arabic information retrieval evaluated their systems on IR performance, using a testing system and a 'test collection' of documents, queries and relevance judgments. This involves substituting different stemmers to see which gives the best results in terms of performance metrics such as Precision, Recall, and F-measure [1]. Such task-specific evaluation makes it impossible to identify typical errors a stemmer would commit. Consequently, this type of evaluation hinders the efforts to devise appropriate solutions and enhancements.

To address this, we use an intrinsic, task-independent evaluation based on correctness, strength and similarity, and apply it to four Arabic stemmers.

This is the first step in tackling current challenges facing Arabic search engines and developing effective search tools that could suit the non-concatenative character of the morphology of Arabic.

2 STEMMER CORRECTNESS EVALUATION

The concept of *stemmer correctness* refers to the capacity of a stemmer to actually merge term variants into a single stem [2]. Because merging processes are prone to error, diverse studies have been carried out to identify the sources of error. In stemming procedures, the inaccuracies appear in the form of under-stemming errors, which occur when words that refer to the same variants are not reduced to the same stem; and over-stemming errors, which occur when words are stemmed incorrectly because they are not actual variants. An assessment approach for stemming algorithms was developed by Paice [3], who evaluates the accuracy of a stemmer by counting the under-stemming and over-stemming errors it commits. His measure provides insights which might help in stemmer and optimization. He introduces three performance evaluation indices: under-stemming index, over-stemming index, and stemming weight. The under-stemming index UI is computed as the proportion of pairs from the sample that are not merged even though they belong to the same group, whereas the over-stemming index OI is computed as the proportion of pairs that belong to different groups among those that are merged to the same stem.

Given a sample of W different words (wordforms) divided into concept groups, he computes the following for each group:

- Desired Merge Total (DMT), given by the following formula:

$$DMT = 0.5n(n-1)$$

- Desired Non-Merge Total (DNT), given by the following formula:

$$DNT = 0.5n(W-n)$$

where n is the number of words in the group.

The sum of the DMT over all groups produces the Global Desired Merge Total (GDMT) and, likewise, the sum of DNT's over all groups yields the Global Desired Non-merge Total (GDNT).

The Unachieved Merge Total (UMT) counts the number of under-stemming errors for each group and is given by the following formula:

$$UMT = 0.5 \sum_{i=1}^s u_i(n - u_i)$$

where s is the number of distinct stems, u_i is the number of instances of each stem. The sum of UMT for all groups yields the Global Unachieved Merge Total (GUMT). The under-stemming index (UI) is given by:

$$UI = \frac{GUMT}{GDMT}$$

The number of over-stemming errors for each group is counted by the Wrongly-Merged Total (WMT) and is given by:

$$WMT = 0.5 \sum_{i=1}^t v_i(n_s - v_i)$$

where t is the number of original groups that share the same stem, n_s is the number of instances of that stem, and v_i is the number of stems for group t . The sum of WMT for all groups is the Global Wrongly Merged Total (GWMT). The over-stemming index (OI) is given by:

$$OI = \frac{GWMT}{GDNT}$$

The *Stemming Weight* (SW), which is a measure of the strength of the stemmer, is calculated by dividing the Over-stemming Index OI by the Under-stemming Index UI. Low SW values indicate a weaker stemmer and higher values indicates a stronger stemmer. A strong stemmer merges a much wider variety of forms, therefore committing many over-stemming errors. A light stemmer fails to merge semantically related words, therefore committing many under-stemming errors. Under-stemming errors tend to decrease the Recall in the IR search, while over-stemming errors will deteriorate Precision. Therefore, correctness metrics facilitate specifying the type of errors made by the stemmers. Consequently, it helps devising appropriate solutions and enhancements with regard to retrieval systems.

3 STEMMER STRENGTH

The degree to which a stemmer changes words that it stems is called *stemmer strength* [4]. Stemmer strength is important because it helps to anticipate recall and precision. There are several ways to measure stemmer strength:

- Number of Words per Conflation Class (WC)—This is the average number of words that are reduced to the same stem. If the conflation of 100 different words resulted in 25 distinct stems, then the mean number of words per conflation class would be 4. Stronger stemmers will have more words per conflation class.
- The *Index Compression Factor* represents the fractional reduction in index size accomplished through the stemming process, the idea being that the heavier the stemmer, the greater the Index Compression Factor. This can be calculated by:
 IC = Index Compression Factor
 N = Number of unique words before stemming
 S = number of unique Stems after stemming
 IC = (N - S)/N
- The mean Levenshtein distance (LD) between words and their stems¹. For example, the Levenshtein distance between “استعان” and “يستعينون” is 4. Our measure will be the average LD for every word in the original sample.

4 INTER-STEMMER SIMILARITY

It is possible to compare two separate stemmers by comparing their outputs. This provides a measure of the similarity (or conversely, the dissimilarity) between the two algorithms. The approach is to take a set of words and apply both algorithms in

¹ The Levenshtein distance between two strings is the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character [5].

turn, thus producing two output lists [1]. Corresponding stems in the two output lists are then compared to give a measure of similarity between the stemmers.

Inter-stemmer similarity could provide valuable information for the designers of IR systems by helping them understand the performance of different stemmers. This type of comparison also helps in developing more efficient stemmers.

$$SSM(U, V) = 100 \left[1 - \left(\frac{\sum_w \left(\frac{LD(U_w - V_w)}{MD(U_w - V_w)} \right)}{N} \right) \right]$$

where

U & V are the stemmers being compared,

N = Number of words in the sample

LD = Levenshtein distance

MD = Maximum Distance

Arabic Stemmers under Consideration

We will now compare the Khoja [5], Light10 [6, 7], Buckwalter [8, 9], and APIR [10] stemming algorithms.

TABLE 1: SUMMARIZATION OF THE FOUR STEMMING ALGORITHMS

Stemmer	type	Algorithm	Lexical resources
Khoja	Root-based	Longest-match affix removal	Yes
Light10	Stem-based	Longest-match affix removal	No
Buck++	Stem-based	Longest-match affix removal	Yes
APIR	Stem-based	Longest-match and dynamic normalization	yes

Khoja's stemmer removes diacritics, stop words, punctuations, and numbers. It then removes the longest suffix and the longest prefix. Finally, it matches the remaining word with verbal and noun patterns, to extract the root. It makes use of several linguistic data files, namely a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words.

A major problem with this type of stemmer is that many word variants are different in meaning, though they originate from one identical root [11].

Larkey's Light10 stemmer is used not to produce the linguistic root of a given Arabic surface form, but to remove the most frequent suffixes and prefixes [11].

Buckwalter developed an Arabic morphological analyzer that returns the possible segmentations of an Arabic word. This analyzer uses three lexicons of possible Arabic prefixes, stems and suffixes, and three compatibility tables to validate the prefix-stem, stem-suffix, and prefix-suffix combinations. It accepts an Arabic word and produces its possible segmentations (transliterated into English characters). It cannot be used directly for stemming, as it provides more than one possible solution for the same word. Thus, we decided to modify it (Buck++) to return the longest stem out of all the stems that might be generated.

Arabic Parsing for Information Retrieval (APIR) was developed by the first author recently. APIR implements the longest-match and dynamic normalization approach. It implements a lexical, or dictionary-based, segmentation which utilizes a lexicon accessed by morphs of the language being analyzed. The input text is scanned (in the right-to-left writing direction) and matches are returned. The longest (or "maximal") match at any given point is returned.

The segmentation part uses the strategy of maximal match segmentation, or "best" segmentation. The maximal match segmentation attempts to minimize the number of words in a sequence of characters by finding the longest matches in the dictionary at each point in the input. APIR employs as-needed normalization to handle internal inflections and boundary distortions. In other words, if there is a mismatch at a point caused by one of the long vowels characters (أ، ي، و، ء) or hamza forms (أ، ء، و، ء)، it will try with another character from each group before starting again.

The lookup dictionary contains only valid Arabic stems without any grammatical or morphological features. Thus, the cost of building this lexical resource and maintaining it is kept minimal.

5 EXPERIMENTAL DATA

The word sample we used in testing the correctness of the four stemmers consisted of 8697 distinct inflectional wordforms collected from Arabic Web sites. We manually categorized them into 1606 conceptual groups. Each group contains only

inflectional wordforms (not derivational variations) and has clear-cut semantic boundaries. As shown in table 8, سقط “to fall (Verb)” is not grouped with سقوط “falling (Noun)”. In the same line, سفارة “embassy” is not grouped with “traveling”, although they are derived from the same root.

TABLE 2: SAMPLES OF CONCEPTUAL GROUPS

Group # n	Group # n+1	Group # n+2	Group # n+3	Group # n+4
تؤيد	السفارات	السفر	سقط	السقوط
تؤيده	السفارة	بالسفر	سقطت	بالسقوط
تؤيدها	بالسفارة	سفر	سقطوا	سقوط
ستؤيده	سفارات	سفرنا	فسقط	سقوطه
سيؤيدونه	سفاراتها	سفرها	فسقطت	سقوطهم
نؤيد	سفارة	للسفر	وسقط	للسقوط
نؤيدك	سفارتها	والسفر	وسقطت	للسقوط
وتؤيد	للسفارات	وسفر	وسقطوا	والسقوط
ونؤيد	للسفارة			وسقوط
ويؤيد	والسفارات			وسقوطهم
يؤيد	والسفارة			

Regarding the wordlist used for inter-similarity and strength evaluation, we have collected 72,000 Arabic words from the Web. This wordlist contains different categories of Arabic words, such as nouns, adjectives, verbs, proper names and transliterated names.

6 STEMMER CORRECTNESS COMPARISONS

A computer program has been written to calculate the UI, OI and SW indices. The program reads the file containing the words sample in addition to the outputs generated by the Khoja, Light10, Buck++ and APIR stemmers. The results are listed in Table 3.

TABLE 3: STEMMING PERFORMANCE INDICES FOR THE FOUR STEMMERS

	UI	OI	SW
Khoja	0.200	0.002286	0.011418
Light10	0.708	0.000236	0.000333
APIR	0.044	0.000025	0.000568
Buck++	0.161	0.000332	0.002051

Light10 has the highest under-stemming errors, followed by Khoja and Buck++. APIR has the lowest under-stemming errors at 0.044. The magnitude of differences is significant between APIR and the other three stemmers. With regard to the over-stemming index, Khoja’s stemmer has the highest value, followed by the Light10 stemmer and then by Buck++. The lowest OI is recorded by the APIR stemmer, with a very significant difference compared to the other three stemmers. These results are graphically shown in Figure 1.

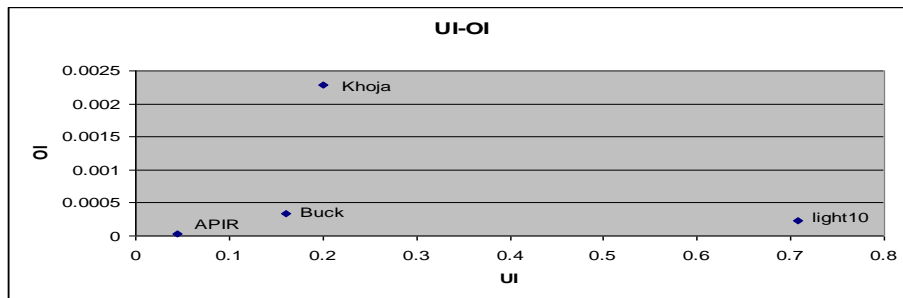


Figure 1: UI vs. OI for the four stemmers

We can say that Light10 commits fewer over-stemming errors compared to Khoja’s and Buck++ stemmers, but leaving many words under-stemmed. On the other hand, Khoja’s stemmer makes fewer under-stemming errors compared to light10 stemmer, but making huge over- stemming errors. This is reflected in the stemmer weight index (SW), SW index of Khoja’s stemmer is very larger compared to the other stemmers, indicating that Khoja’s stemmer is the strongest one. What is interesting is the SW of APIR. Its value is less than Khoja and Buck++ but more than light10, indicating that it makes less over-stemming error and less under-stemming errors (more ideal stemmer). In summary the order of stemmer strength is:

Khoja> Buck++>APIR>light10

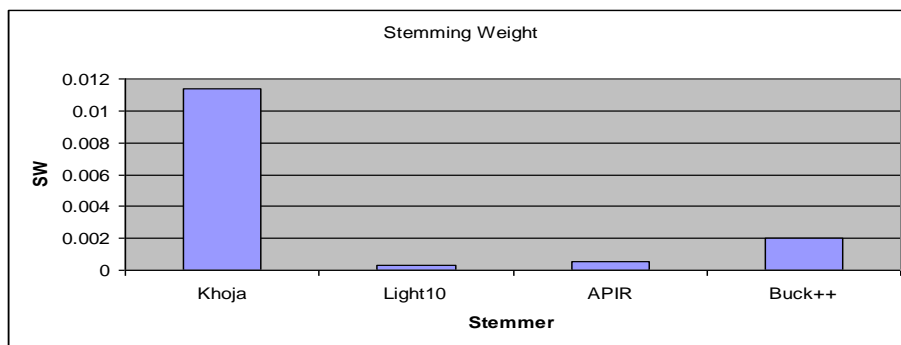


Figure 2: Stemmer strength

7 STRENGTH COMPARISONS

In this section, we analyze the strengths of the four stemmers using 3 measures: Levenshtein Distance, Words per Conflation Class, and Index Compression.

TABLE 4: RESULTS OF STEMMER STRENGTH MEASURES

Stemmer	LD	WC	IC
Light10	1.59	2.14	0.53
APIR	1.89	4.30	0.76
Buck++	1.95	4.48	0.77
Khoja	2.84	7.17	0.86

The three metrics listed in table 4 are consistent in ordering the relative strengths of the stemmers. Based on these metrics, we found that Khoja is the strongest stemmer. We also noticed that both Buck++ and APIR are considerably weaker than Khoja’s stemmer. This is because Khoja’s stemmer extracts roots, while Buck++ and APIR are stem-based algorithms. Certainly, on average, the distance between a word and its stem is less than the distance between a word and its root. Light10 stemmer is a weak stemmer compared to other three stemmers.

Each measure places the stemmers in the following order:

Khoja> Buck++>APIR>light10

These results correspond exactly with the Stemming Weight results obtained using correctness measures.

8 INTER-STEMMER SIMILARITY COMPARISONS

We have applied the wordlist containing 72,000 entries to the four stemmers. We then calculated the average distance for all pairs of stems. The results are listed in table 5 for each pair.

TABLE 5: SIMILARITY MEASURES

Pairs	Inter-stemmer similarity	Percentage of same stems
APIR-Buck++	91.23	68.41
Light10-Buck++	81.6	40.43
APIR-Light10	81.47	39.17
Khoja-Buck++	69.11	20.07

APIR-Khoja	66.10	15.06
Light10-Khoja	64.01	14.63

The results suggest that the inter-similarity pairings from most similar to least similar are: APIR-Buck++, Light10-Buck++, APIR-Light10, Khoja-Buck++, APIR-Khoja and Light10-Khoja.

These results are validated by stemmer strength evaluation. We have seen that APIR and Buck++ are closer to each other in terms of strength metrics. This is also valid for the inter-similarity metric, as the APIR-Buck++ pair has the highest relatedness. We also notice that Light10 is more similar to both Buck++ and APIR than to Khoja, which is also apparent in the strength measures. The lowest similarity is detected in the pairs involving Khoja's stemmer which has a very high strength compared to other stemmers.

Hence, the inter-stemmer similarity measure is in total agreement with the results obtained from strength measures. However, the inference of similarity pairings from the correctness indices discussed above is not straightforward. In terms of under-stemming errors, APIR is more similar to Buck++, then to Khoja and finally to Light10. With regard to over-stemming errors, APIR similarity with Light10 is higher than with Buck++, and its similarity with Khoja is the lowest.

To demonstrate this, we will try to find the Correctness Similarity metric (CSM). The correctness similarity between two stemmers can be calculated by finding the difference of UI ratio and OI ratio of the two stemmers. For identical stemmers, the CSM would be 0. The CSM is given by the following formula:

$$CSM(U, V) = \left(\frac{UI_u}{UI_v} - \frac{OI_u}{OI_v} \right)$$

Where

U & V are the Stemmers being compared,

UI = Under-stemming index

OI = Over-stemming index

TABLE 6: CORRECTNESS SIMILARITY RESULTS

Pairs	Correctness Similarity
APIR-Buck++	9.6
Light10-Buck++	3
APIR-Light10	6.5
Khoja-Buck++	5.6
APIR-Khoja	86.9
Light10-Khoja	6.15

Table 6 summarizes the results obtained and compares them with the distance-based similarity. We observe that there is no agreement between the two lists. For example, the APIR-Buck++ pair is very similar in terms of distance, but not similar in terms of correctness. Hence, we conclude that, as in the case of stemmer strength, inter-stemmer similarity is not directly related to correctness. Thus, one could have two stemmers which are very similar and yet which are virtually different in their ability to conflate related words [1].

TABLE 7: CORRECTNESS-BASED VS. DISTANCE-BASED SIMILARITY

Correctness Similarity (high to low)	Distance Similarity (high to low)
Light10-Buck++	APIR-Buck++
Khoja-Buck++	Light10-Buck++
Light10-Khoja	APIR-Light10
APIR-Light10	Khoja-Buck++
APIR-Buck++	APIR-Khoja
APIR-Khoja	Light10-Khoja

9 CONCLUSIONS

In this paper we evaluated the correctness, strength of four stemming algorithms (Khoja, Light10, Buck++ and APIR), and their mutual similarities.

Stemmer correctness, that is, the ability of the stemmer to conflate related words accurately is important, because it provides insight into the types of errors stemming algorithms commit, and helps devise solutions and enhancements with regard to retrieval experimentation. Based on the number of under- and over-stemming errors, APIR outperforms other stemmers significantly.

Stemmer strength measures the amount of alteration on wordlist a stemmer can make. Using stemmer strength is useful in predicting index size, recall and precision in IR systems. We found that all metrics are consistent in ranking the relative strength of the four stemmers. Remarkably, this ranking corresponds precisely with the Stemmer Weight (SW) results.

These evaluation methods are not alternative but complementary, and the results presented provide a baseline for further enhancement and development.

REFERENCES

- [1] R. Hooper and C. Paice, "Evaluation Techniques," vol. 2010: Lancaster University, 2006.
- [2] C. Galvez and F. d. Moya-Anegón, "An evaluation of conflation accuracy using finite-state transducers," *Journal of Documentation*, vol. 62, pp. 328-349, 2006.
- [3] D. P. Chris, "An evaluation method for stemming algorithms," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Dublin, Ireland: Springer-Verlag New York, Inc., 1994.
- [4] B. F. William and J. F. Christopher, "Strength and similarity of affix removal stemming algorithms," *SIGIR Forum*, vol. 37, pp. 26-30, 2003.
- [5] S. Khoja and R. Garside, "Stemming arabic text." Lancaster, UK. Computer Science Department, Lancaster University, 1999.
- [6] L. S. Larkey and L. Ballesteros, "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis," presented at SIGIR 2002, 2002.
- [7] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Light Stemming for Arabic Information Retrieval " in *Arabic Computational Morphology*, A. Soudi, A. v. d. Bosch, and G. Neumann, Eds.: Springer Netherlands, 2007.
- [8] T. Buckwalter, "Arabic lexicography," QAMUS, 2002.
- [9] LDC, "Buckwalter Morphological Analyzer Version 1.0," Linguistic Data Consortium, 2002.
- [10] D. Daoud and H. Qais, "Stemming Arabic using Longest-Match and Dynamic Normalization," presented at Arabic Language Technology International Conference (ALTIC) 2011, Bibliotheca Alexandrina (B.A.), Alexandria, Egypt, 2011.
- [11] T. Naglaa, "Stemming the Qur'an," in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*. Geneva, Switzerland: Association for Computational Linguistics, 2004.

Tapping into the Power of Automatic Scoring

Wael H. Gomaa^{*1}, Aly A. Fahmy^{*2}

**Computer Science Department, Faculty of Computers & Information, Cairo University*

Giza, Cairo, Egypt

¹wael.gomaa@gmail.com

²a.fahmy@fci-cu.edu.eg

Abstract - The increasing number of students and tests made the process of answer assessment a night mare. Automatic Scoring (AS) reduces time, provides evaluation consistency and standardization. AS systems are wide enough to cover all types of student's conducted response writing, speech and mathematics. This paper presents a comprehensive survey on AS technology and its applications.

1 INTRODUCTION

This paper discusses automated scoring (AS) technology which refers to a large collection of grading approaches that differ depending on the constructed-response (CR) being posed and the expected answer. AS offers many advantages as increasing scoring consistency, introducing varied high-stakes assessments, reducing processing time and keeping the meaning of "Standardization" by applying the same criteria to all the responses; . In other words AES provides benefits to all assessment tasks' components students, evaluators and testing operation.

Although the aim of AS systems is achieving a high correlation between the grades of both human and machine, it is important to know that a machine grading and human grading for students answers differs. Generally speaking there are two grading methods; first method depends on exact-matching between students' answer CR and the saved correct answer(s), second method depends on extracting and analyzing different features from student answers to generate the automatic score.

Current AS research deals with students' CR for writing, speaking and mathematical responses; writing assessment includes essays and short answer grading, speaking assessment includes low and high entropy spoken responses, mathematical assessments include textual, numeric or graphical responses.

This paper focuses on the methodologies and results of different applications for the major AS developers like: Educational testing Service (ETS), Pearson Knowledge technologies (PKT) and Vantage Learning.

The paper is organized into four main sections: Automatic Essay Scoring (AES), Short Answer Scoring, Speech Scoring and Math Scoring.

2 AUTOMATIC ESSAY SCORING (AES)

Automated essay scoring (AES) is defined as the computer technology that evaluates and scores written works [1]. AES is also known as automated essay evaluation, automated essay assessments and automated writing scoring.

Most AES work is designed for English language where only few studies were designed to support other languages like Japanese, Hebrew and Bahasa Malay [2].

AES goes through the same steps of any supervised algorithm; training, features extraction and finally testing. The steps of building AES model are simply as follows [3]; first a training sample of hundreds essay responses are assessed by experts (raters), this training sample is then examined by computers to identify and extract a set of text features and weights to produce a model that can be used to predict the human rating, this model is validated by comparing the results manually obtained by human raters and the computerized model, finally when the scoring model gives satisfactory result, new responses can be automatically graded.

There are two main approaches to create AES models either using brute-empirical methods or hybrid methods [3]. The first approach uses a large variety of linguistic features that have no direct relation to writing theory while models based on hybrid methods have a direct relation to a theoretically derived conception of the characteristics of good writing.

AES Systems:

A. *Project Essay Grader (PEG)* is the leading AES system in the history of automatic assessment. It depends on proxy measures to predict essays intrinsic quality. Proxies refer to a particular writing construct such as average word length, average sentences length, and count of other textual units [3, 4]. It used a statistical procedure to produce feature weights which is simple multiple regression. The original version of PEG was created by Ellis Page in 1966 at the University of Connecticut [5]. In 1990's an enhanced version of the system that used Natural language Processing (NLP) tools was released. That version presented NLP tools as syntactic analysis which focuses on grammar checkers and part of speech (POS) tagging.

B. Intelligent Essay Assessor (IEA) focuses mainly on the evaluation of content. IEA scores essays using LSA [6,7] which is a semantic text analysis method that can be defined as “a statistical model of word usage that permits comparisons of the semantic similarity between pieces of textual information” [8]. IEA combines the LSA method with informational database that contains textbook material, sample essays or other sources rich in semantic to train computers. This combination requires fewer human scored essays in the IEA training sample as scoring is accomplished based on semantic analysis rather than statistical models [9]. IEA was originally developed at the University of Colorado in 1997 and has recently been purchased by Person Knowledge Technology (PKT). It is a back-end service that uses the KAT™ engine and a customer's Web interface to evaluate essays as reliably as skilled human readers.

IEA has many advantages over other essay scoring systems as it provides an overall evaluation and feedback on spelling and grammar errors. It also has built in detectors for highly unusual essays. Besides operating as a Web-based service, IEA can be customized as well as licensed with an optional user management system. IEA's underlying KAT engine is highly reliable as it was used for scoring over a million essays ranging from middle school to medical school, in a variety of content areas.

C. Intellimetric was developed by Vantage Learning Technology in 1997 as a part of a web-based portfolio administration system called MyAccess!. Intellimetric is the first artificially intelligent based AES that combines the tools of Natural Language Processing (NLP) and statistical technologies in essay scoring. It can be referred to as a learning engine that internalized the "pooled wisdom" or "brained based" of expert human evaluators [10]. Intellimetric uses a model that contains optimal set of predictors and weights that are defined by extracting more than 400 features from student answers, in addition to, a training set that consists of semantic, syntactic and discourse related features.

The basic five dimensions scores underlying the IntelliMetric system are Content, Creativity, Style, Mechanics and Organization. Intellimetric uses word nets based on statistical semantic text similar to LSA which is Latent Semantic Dimension (LSD). LSD features are described in five broad categories. The first is focus and unity which cares of cohesiveness and consistency in purpose and main ideas in an essay. The second category is development and elaboration which indicates the breadth of the content and the supporting ideas, i.e. vocabulary, elaboration, word choice and concepts. The third category cares with essay organization and structure as the logic of discourse including transitional fluidity and relationships among parts of response. The fourth category of sentence structure focuses on sentence complexity and variety such as syntactic variety, sentence complexity. Finally, the fifth category is mechanics and conventions which analyze the essay's conformance of English language rules as grammar, spelling, capitalization, sentence completeness, and punctuation [10].

D. E-rater is the AES system developed by Educational Testing Service (ETS). E-rater is well known for scoring predictions that are comparable to human reader scores in addition to its capability to automatically detect off-topic responses [13, 14, 15, 16,17]. E-rater is currently used for:

- Scoring essays submitted to ETS's writing instruction application.
- Scoring the Graduate Management Admission Test Analytical Writing Assessment (GMAT® AWA).
- The scoring application of Criterion Online Essay Evaluation Service which is a web-based commercial essay evaluation system. In this application the e-rater engine simply scores the essay by extracting linguistically-based features from the essay and uses a statistical model to relate these features to overall writing quality. The essay is given a score of 1 to 6 where 1 is the lowest score and 6 is the highest score [11, 12].

E-rater version 1.3 applied stepwise linear regression to a training sample of essays written on the same topic that had been scored by human readers in order to compute more than 50 linguistically based feature scores that can be of a great help in the prediction of essay scores [12].

E-rater version 2 is composed of up to 12 essay scoring features associated with five areas of analysis; first Errors in Grammar, Usage, Mechanics, and Style. Second is Organization and Development. Third is Lexical Complexity. Fourth is Prompt-Specific Vocabulary Usage and finally is Essay Length. E-rater includes other essay scoring features related to vocabulary, content appropriateness, organization and development.

E. C-rater™ has been developed by ETS and is well known for high scoring accuracy for written responses as it has been validated on responses from multiple testing programs in many different content areas, including science, reading comprehension and history [18,19].

C-rater's technology uses "bag of words approach" in which deep natural language processing is used to assess whether a student response contains text which could be considered a paraphrase of the concepts listed in the rubric for an item. This approach contrasts with other methods for scoring student responses as LSA (Latent Semantic Analysis) that are primarily based on the type of words used rather than how they are put together to form higher-level meaning units. C-rater engine applies a sequence of NLP steps [19,20], including:

- Correcting students' spelling.
- Determining the grammatical structure of each sentence.
- Resolving pronoun reference.

- Analyzing paraphrases in student responses.

The main advantage of c-rater over other AES engines is the deep linguistic analysis of student responses which ensures that the scoring process will not be misled by responses that use the right words in the wrong context.

AES Applications' Results:

The following table represents applications' results achieved in terms of test, sample size of scored essays, human-human correlation and human-computer correlation.

TABLE 1
AES APPLICATIONS' RESULTS

System	Test	Sample size	Human-Human r	Human- Computer r
PEG (1997)	GRE	497	.75	.74-.75
PEG (2002)	English placement test	386	.71	.83
IntelliMetric (2001)	k-12 norm- referenced test	102	.84	.82
IEA (1997)	GMAT	188	.83	.80
IEA (1999)	GMAT	1363	.86-.87	.86
IEA (2011)	High School Writing	635	0.91	0.91
e-rater (1998)	GMAT	500-1000	.82-.89	.79-.87
e-rater (2006)	GMAT – TOEFL	7575	.93	.93
e-rater (2011)	GRE- TOEFL	>5000	.95	.97

3 SHORT ANSWER GRADING

Short Answer Grading systems are easy to implement as they are meant to assess student's content knowledge and skills; in opposite to Essay grading systems that assess student's writing ability and require sophisticated text understanding and analysis. Short Answer Grading systems require student to respond with short text demonstrating his or her understanding of key concepts in a certain domain. Automatic Answer Grading system is one which automatically assigns a grade to an answer provided by a student through a comparison with one or more correct answers. In the past most short answer grading systems depended on manual answers patterns selection where a matched pattern indicates right answer, other systems require annotated corpus to select answer patterns in semi-automatically way [21, 22].

Automatic Grading systems are easy to implement for Questions like Multiple Choice, True-False, Matching and Fill-in-the-blank.

Short Answer Grading Systems:

A. *Oxford-UCLES* [21] this system uses a set of keywords, synonyms and window searching for pattern selection. The system was upgraded [22] to compare several machine learning approaches like decision tree learning, Bayesian learning and inductive logic programming.

The application was evaluated by experimenting 260 answers for each of the 9 questions taken from a UCLES GCSE biology exam. The marks for these questions ranged from 1 to 4. The training set contained 200 marked answers and 60 unmarked answers were used as the testing set. When the application depended on handcrafted pattern selection the average percentage agreement between the automatic system and the marks assigned by human examiner was 84%.

When comparing to the results of the application that depended on machine learning techniques; hand crafted approach showed higher accuracy.

B. *C-rater* is an automated scoring system that uses morphological analysis, synonyms, predicate argument structure and pronominal reference [23] to evaluate responses to content-based short answer questions.

C-rater has been evaluated in two large-scale assessment programs [23]. The first was the National Assessment of Educational Progress (NAEP) Math Online Project. C-rater was used to evaluate written explanations of the reasoning behind particular solutions to some mathematical problems. Five questions were used in the evaluation process.

The second program was the online scoring and administration of Indiana's English 11 End of Course Assessment pilot study. In this case, C-rater was required to evaluate seven reading comprehension questions. The answers to these questions were more open-ended than those to the questions in NAEP Math Online Project. In the NAEP assessment, the average length of the responses was 1.2 sentences or 15 words. Between 245 and 250 randomly chosen student responses were scored by two human judges and by C-rater. The average agreement rate between C-rater and the first human judge was 84.4% while between C-rater

and the second human judge it was **83.6%**. The average agreement rate between the two human judges was **90.8%**. This means that C-rater's performance was encouraging in the case of the NAEP assessment.

C. Automark is a software system that employs NLP techniques to perform computerized marking of free-text answer to open-ended questions [24, 25]. It uses Information Extraction techniques to extract the concept or meaning behind free text. Its marking is primarily based on content analysis but certain style features may also be considered. The marking process goes through 4 stages. First, student answer is pre-processed to be standardized in terms of punctuation and spelling and to ensure that the system is tolerant of errors in typing, spelling and syntax. Second sentence analyzer identifies the main syntactic constituents of the text and how they are related. Third pattern-matching module searches for matches between the marking scheme templates and the syntactic constituents of the student text. Finally, the feedback module processes the result of the pattern match and feedback is typically provided as a mark, but more specific feedback is claimed to be possible [25]. Automark has been tested on National Curriculum Assessment of Science for eleven years old pupils. The form of response was: single word generation, single value generation, generation of a short explanatory sentence, description of a pattern in data. The correlation achieved ranged between **93%** and **96%**.

D. Text similarity approach is a grading system in which grade is assigned based on comparing several relatedness measures between the student answer and the instructor answer [26, 27]. Several relatedness measures are used including knowledge-based through Shortest path, Leacock & Chodorow, Lesk, Wu & Palmer, Resnik, Lin, Jiang & Conrath, Hirst & St-Onge algorithms and corpus based measures through LSA and ESA techniques. The best results was obtained with a corpus-based measure using Wikipedia combined with a "relevance feedback" approach that iteratively augments the instructor answer by integrating the student answers that receive the highest grades.

4 SPEECH SCORING

Automated scoring of speech is very similar to automated essay scoring. First, language related features are extracted, and then a scoring model is used to compute a score based on a combination of these features. Automated Scoring of essays and speech differs in two main points first: speech scoring requires additional programming to generate word hypotheses from the digitized student's speech response to an item prompt. Second speech testing is generally done for non-native speakers. Speech scoring tasks are classified in two basic categories: low-entropy and high-entropy tasks. Low-entropy tasks scores responses that are fairly predictable as oral reading from a printed passage, repeating an orally presented stimulus, giving an answer to a highly constrained factual question and describing a simple picture. In contrast high-entropy tasks produce unrestricted, spontaneous speech.

Speech Scoring Systems:

A. ETS's SpeechRater engine is the only spoken response scoring application that is used to score spontaneous responses, in which the range of valid responses is open ended rather than narrowly determined by the item stimulus. Test takers preparing to take the TOEFL test have had their responses scored by the SpeechRater engine as part of the TOEFL Practice Online test since 2006. Competing capabilities focus on assessing low-level aspects of speech production such as pronunciation by using restricted tasks in order to increase reliability. The SpeechRater engine, by contrast, is based on a broad conception of the construct of English speaking proficiency, encompassing aspects of speech delivery (such as pronunciation and fluency), grammatical facility and higher-level abilities related to topical coherence and the progression of ideas [26,27].

The SpeechRater engine processes each response with an automated speech recognition system specially adapted for use with nonnative English. Based on the output of this system, natural language processing is used to calculate a set of features that define a "profile" of the speech on a number of linguistic dimensions, including fluency, pronunciation, vocabulary usage and prosody. A model of speaking proficiency is then applied to these features in order to assign a final score to the response. While the structure of this model is informed by content experts, it is also trained on a database of previously observed responses scored by human raters, in order to ensure that SpeechRater's scoring emulates human scoring as closely as possible. Furthermore, if the response is found to be unscorable due to audio quality or other issues, the SpeechRater engine can set it aside for special processing [28,29].

ETS's research agenda related to automated scoring of speech includes the development of more extensive Natural Language Processing (NLP) features to represent grammatical competencies and the discourse structure of spoken responses. The core capability is also being extended to apply across a range of item types used in different assessments of English proficiency, including a range of options from very restricted item types (such as passage read-alouds), through less restrictive items (such as summarization tasks), to fully open-ended items.

B. PKT's Versant is an automated spoken language test that can be easily taken over a phone or computer by large groups of candidates. Tests are automatically scored within minutes and provide both an overall score and sub-skill scores [30]. The

Versant tests have helped corporations, government agencies, universities, and schools accurately and quickly measure spoken English, Spanish, or Arabic skills [31] for screening and training purposes in over 100 countries around the world.

The Versant testing system automatically scores responses to many different item tasks. In the Versant Speaking tests, these may include: reading aloud, repeating sentences, building sentences, giving short answers to questions, retelling brief stories, response selection, conversations, and passage comprehension. In the Versant Writing test, item tasks include: typing, completing sentences, dictation, reconstructing passages, and writing e-mails. For some tasks, such as Reading and Repeats, there is exactly one correct word sequence expected for each response. In other tasks, items can have multiple correct answers. All test items have undergone extensive pre-testing on diverse samples of native and non-native speakers at a wide range of ability levels [30].

C. SRI International's EduSpeak system is a software development toolkit that enables developers of interactive language education software to use state-of-the-art speech recognition and pronunciation scoring technology [32]. Automatic pronunciation scoring allows the computer to provide feedback on the overall quality of pronunciation and to point to specific production problems. We review our approach to pronunciation scoring, where our aim is to estimate the grade that a human expert would assign to the pronunciation quality of a paragraph or a phrase. Using databases of nonnative speech and corresponding human ratings at the sentence level, we evaluate different machine scores that can be used as predictor variables to estimate pronunciation quality. For more specific feedback on pronunciation, the EduSpeak toolkit supports a phone-level mispronunciation detection functionality that automatically flags specific phone segments that have been mispronounced. Phone-level information makes it possible to provide the student with feedback about specific pronunciation mistakes. Two approaches to mispronunciation detection were evaluated in a phonetically transcribed database of **130,000** phones uttered in continuous speech sentences by **206** nonnative speakers. Results show that classification error of the best system, for the phones that can be reliably transcribed, is only slightly higher than the average pair wise disagreement between the human transcribers [32].

5 MATHEMATICS SCORING

In the area of mathematics, the performance of automated scoring systems is typically quite robust when the response format is constrained. The types of mathematics item responses that can be scored by automated systems include mathematical equations or expressions, two-dimensional geometric figures, linear, broken-line or curvilinear plots, bar graphs, and numeric entry. The field has experienced at least eight years of advances in these systems since they were first deployed in consequential statewide assessments, and it is reasonable to expect these systems to perform with high accuracy. This enables the use of these systems without additional oversight by human raters. Automatic scoring of freehand graphic responses and handwritten expressions achieves lower accuracies. For the more constrained response types, the most notable limitation is that automated scoring assumes computer test delivery and data capture, which in turn may require an equation editor or graphing interface that students can use comfortably.

Mathematics Scoring Systems:

A. ETS's m-rater scoring engine is used for scoring open-ended mathematical responses, such as those which take the form of mathematical expressions, equations or graphs. Dating from the late 1990s, the m-rater scoring engine ranks among the ETS automated scoring capabilities with the longest development history and demonstrates very strong agreement with human ratings (as one would expect in the mathematics domain).

The m-rater scoring engine evaluates the correctness of a mathematical expression based on numerical equivalence, enabling it to identify expressions equivalent to the key no matter what form they are found in, and to assign credit as appropriate. For instance, partial credit may be assigned if a linear equation was supposed to be provided in slope-intercept form, but it was instead provided in a different, equivalent form. Scoring of mathematical responses based on string matching or text-based patterns is much more limited and error-prone than the m-rater scoring engine's capabilities for establishing true numerical equivalence [34,35].

Similarly, graph items can be scored based on a key which specifies constraints on the response entered with the graph editor. For some items, many different graphs may constitute valid answers, and the m-rater scoring engine can allow all of these variants to be scored using an elegant specification of the key.

Of course many math items are written to elicit short, text-based responses and may be more suitable for the c-rater™ engine. Written responses with embedded equations can even be handled using a hybrid of the m-rater and c-rater scoring engines [34].

B. Pearson's MathQuery [35,36] is a web-based environment that exercises and assesses critical thinking skills in math. These skills are best measured by multistep and real-world problems that can be solved more than one way and that can have multiple

valid solutions that are not equivalent. MathQuery brings together technologies that display high-quality math notation and that allow student input of well-formed math responses. Criterion-based assessment algorithms automate important aspects of human scoring that go beyond numerical equivalence scoring. MathQuery generates and scores classes of algebraic and graphic problems using item schemas.

Note that this kind of math problem has multiple paths to the correct answer. In order to provide formative feedback and/or give partial credit, MathQuery analyzes the sequence of steps or the path to the solution. For mathematical expressions, MathQuery offers an equation editor that can be customized for different grade levels and content areas, so that pre-algebra students can easily express fractions, but are not overwhelmed by the functionality and symbols needed for calculus. In addition, the equation editor can correct input errors during response construction and if errors are not caught during input, MathQuery's assessment engine can accommodate input errors during grading by adapting the assessment criteria to the unexpected input.

6 CONCLUSION

The diversity of Automatic Scoring fields; writing, speech and mathematics is a great advantage for evaluators. In this paper we introduced different systems for all automatic scoring fields. Systems' accuracy is the correlation between human grading and system grading. As long as there is a difference between automatic grading and human grading the accuracy issue is a good point of research.

REFERENCES

- [1] Shermis, M. D. & Burstein, J., "Automated Essay Scoring: A Cross Disciplinary Perspective", Mahwah, NJ: Lawrence Erlbaum Associates, 2003.
- [2] Shermis, M. D., Burstein, J., Higgins, D., & Zechner, K., "Automated Essay Scoring: Writing Assessment and Instruction", In B. McGaw, P. L. Peterson & E. L. Baker (Eds.), *International Encyclopedia of Education*. Maryland Heights, MO: Elsevier B.V., 2010.
- [3] Ben-Simon, A., & Bennett, R. E., "Toward theoretically meaningful automated essay scoring", the annual meeting of the National Council of Measurement in Education, San Francisco, CA., 2006.
- [4] Page, E. B., "The imminence of grading essays by computer," *Phi Delta Kappan*, 48, 238-243, 1996.
- [5] Semire Dikli, "Automated Essay Scoring", *Turkish Online Journal of Distance Education-TOJDE*, ISSN 1302-6488 Volume: 7 Number: 1 Article: 5, 2006.
- [6] Landauer, T. K., Foltz, P. W., & Laham, D., "Introduction to latent semantic analysis", *Discourse Processes*, 25, 259-284, 1998.
- [7] Landauer, T. K., Laham, D., & Foltz, P. W. , "Automated scoring and annotation of essays with the Intelligent Essay Assessor", In M. D. Shermis & J. Burstein (Eds.), "Automated essay scoring: A cross-disciplinary perspective," (pp. 87-112), Mahwah, NJ: Lawrence Erlbaum Associates, Inc, 2003.
- [8] Foltz, P. W. , "Latent Semantic Analysis for text-based research", *Behaviour Research Methods, Instruments and Computers*, 28(2), 197-202, 1996.
- [9] Warschauer, M., & Ware, P., "Automated writing evaluation: Defining the classroom research agenda", *Language Teaching Research*, 10(2), 157-180, 2006.
- [10] Elliot, S. "Intellimetric: From here to validity", In M. D. Shermis & J. Burstein (Eds.), "Automated essay scoring: A cross disciplinary perspective", (pp. 71-86). Mahwah, NJ: Lawrence Erlbaum Associates, Inc, 2003.
- [11] Burstein, J., Chodorow, M., & Leacock, C., "CriterionTM: Online essay evaluation: An application for automated evaluation of student essays", In J. Riedl & R. Hill (Eds.), *Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence*, Acapulco, Mexico (pp. 3-10). Menlo Park, CA: AAAI Press, 2003.
- [12] Attali, Y., & Burstein, J., "Automated Essay Scoring With e-rater V.2", *Journal of Technology, Learning and Assessment* 4(3), Available from <http://www.jtla.org>, 2006.
- [13] Tetreault, Joel and Foster, Jennifer and Chodorow, Martin, "Using parse features for preposition selection and error detection", In: *ACL 2010 - 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010.
- [14] J. Tetreault, E. Filatova, & M. Chodorow, "Rethinking Grammatical Error Annotation and Evaluation with the Amazon Mechanical Turk", *NAACL-HLT: Proceedings of the 5th Workshop on Building Educational Applications (BEA-5)* Association for Computational Linguistics, 2010.
- [15] J. Burstein & M. Chodorow, "Progress and New Directions in Technology for Automated Essay Evaluation", *The Oxford Handbook of Applied Linguistics*, 2nd Edition, pp. 487-497 Editor: R. Kaplan Oxford University Press, 2010.
- [16] A. Louis & D. Higgins, "Unsupervised Prompt Expansion for Off-Topic Essay Detection", *Proceedings of the Workshop on Building Educational Applications, HLT-NAACL*, Association for Computational Linguistics, 2010.
- [17] J. Burstein, J. Tetreault, & S. Andreyev, "Using Entity-Based Features to Model Coherence in Student Essays", *Human language technologies: The Annual Conference of the North American Chapter of the ACL*, pp. 681-684 Association for Computational Linguistics, 2010.

- [18] J. Z. Sukkarieh & S. Stoyanchev, "Automating Model Building in c-rater", Proceedings of TextInfer: The ACL/IJCNLP Workshop on Applied Textual Inference, pp. 61–69, Association for Computational Linguistics, 2009.
- [19] J. Z. Sukkarieh & J. Blackmore, "c-rater: Automatic Content Scoring for Short Constructed Responses", Proceedings of the 22nd International FLAIRS Conference Association for the Advancement of Artificial Intelligence, 2009.
- [20] J. Z. Sukkarieh & E. Bolge, "Building a Textual Entailment Suite for Evaluating Content Scoring Technologies", Proceedings of the Seventh Conference on International Language Resources and Evaluation, pp. 3149–3156, European Language Resources Association, 2010.
- [21] J.Z. Sukkarieh, S.G. Pulman, and N. Raikes. , "Auto-Marking 2: An Update on the UCLES-Oxford University research into using Computational Linguistics to Score Short, Free Text Responses", International Association of Educational Assessment, Philadelphia, 2004.
- [22] S.G. Pulman and J.Z. Sukkarieh., "Automatic Short Answer Marking", ACL WS Bldg Ed Apps using NLP, 2005.
- [23] C. Leacock and M. Chodorow., "C-rater: Automated Scoring of Short Answer Questions", Computers and the Humanities, 37(4):389–405, 2003.
- [24] T. Mitchell, T. Russel, P. Broomhead and N. Aldridge, "Towards robust computerized marking of free-text responses", Proceedings of the Sixth International Computer Assisted Assessment Conference, Loughborough, UK: Loughborough University, 2002.
- [25] S. Valenti, F. Neri and A. Cucchiarelli, "An overview of current research on automated essay grading", Journal of Information Technology Education, vol. 2, pp. 319–330, 2003.
- [26] K. Zechner & X. Xi , "Towards Automatic Scoring of a Test of Spoken Language with Heterogeneous Task Types" , Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications, pp. 98–106, Association for Computational Linguistics, 2008.
- [27] K. Zechner, D. Higgins, X. Xi, & D. Williamson, "Automatic Scoring of Non-Native Spontaneous Speech in Tests of Spoken English", Speech Communication, Educational Testing Service, Automated Scoring and NLP, Rosedale Road, MS 11 R, Princeton, NJ 08541, USA , Vol. 51, No. 10, pp. 883–895, 2009.
- [28] Derrick Higgins, Xiaoming Xi, Klaus Zechner, and David Williamson, "A three-stage approach to the automated scoring of spontaneous spoken responses", Computer Speech and Language, 25(2), pp.282-306, 2011.
- [29] L. Chen, J. Tetreault, & X. Xi, "Towards Using Structural Events to Assess Non-Native Speech" , NAACL-HLT: Proceedings of the 5th Workshop on Building, Educational Applications (BEA-5) Association for Computational Linguistics, 2010.
- [30] Bernstein, J., van Moere, A., and Cheng, J., "Validating automated speaking tests", Language Testing, July, 27, 355-377, 2010.
- [31] Bernstein, J., Suzuki, M., Cheng, J. and Pado, U., "Evaluating diglossic aspects of an automated test of spoken modern standard Arabic", ISCA International Workshop on Speech and Language Technology in Education ,SLaTE, 2009.
- [32] H. Franco, H. Bratt, R. Rossier, V. R. Gadde, E. Shriberg, V. Abrash, and K. Precoda, "EduSpeak: a speech recognition and pronunciation scoring toolkit for computer-aided language learning applications", Language Testing, vol. 27, no. 3, p. 401, 2010.
- [33] R.E. Bennett, " Automated Scoring of Constructed-Response Literacy and Mathematics Items", White Paper, Publisher: Arabella Philanthropic Advisors, 2011.
- [34] B. Sandene, R. E. Bennett, J. Braswell, & A.Oranje, "Online Assessment in Mathematics and Writing: Reports from the NAEP Technology-based Assessment Project", (NCES 2005–457) U.S. Department of Education, National Center for Education Statistics, 2005.
- [35] Deland, D., "An RIA Approach to Web Mathematics", presented at AMS/MAA Joint Mathematics Meeting, San Francisco, January 2010.
- [36] Dooley, S. S., "MathEX: A Direct-Manipulation Structural Editor for Compound XML Documents", In Proceedings of the Mathematical User-Interfaces Workshop, Schloss Hagenberg, Linz, Austria, 2007.

أثر تجاور صوتي الفعل الثلاثي المضعف في بابهِ الصرفي: دراسة لغوية حاسوبية على الأصوات الذلقية (ر- ل- ن)

أ. د. وفاء كامل فايد

أستاذة اللغويات – كلية الآداب بجامعة القاهرة

مقدمة :

خلصت دراستي عن (تراكب الأصوات في الفعل الثلاثي الصحيح)⁽¹⁾ إلى عدد من القواعد التي تحكم تنافر الأصوات العربية وتآلفها؛ مما يشير إلى أن وراء السلوك اللغوي التلقائي للعربية نظاماً داخلياً يحدد النماذج المقبولة ويميزها عن النماذج غير المقبولة، وهو ما أتاح للحس العربي أن يقبل منها ما هو جدير بالقبول، ويعرض عن سواه.

وفكرت في أن أتجه إلى الفعل الثلاثي المضعف المجرد، الذي يتكون من صوتين، مثل الفعلين (ردّ، رِقّ)، فنحن نرى أن الفعل (ردّ) مضارعه: يرُدّ- على باب نصر ينصُر- والفعل (رِقّ) مضارعه: يرقّ، على باب ضرب يضرب؛ لأرى هل يكون لصوتي هذا النوع من الأفعال أثر في اتجاه الفعل للتصرف على باب صرفي دون غيره؟ وهل يمكن معرفة أثر تجاور الصوتين على الباب الصرفي؟ وهل يكون سلوك الفعل المضعف على باب صرفي بعينه راجعاً إلى سيطرة هذا النموذج المختزن في المعجم الذهني العربي؟ أسئلة راحت تلح على تفكيري فحاولت البحث عن إجابة لها.

وبدأت بتتبع هذا النوع من الأفعال؛ لألحظ هل يؤثر تجاور صوتيه في تصرف مضارعه على باب صرفي بعينه، فوجدت أن بعض الأفعال يتحد فيها الصوت الأول (الفاء) ويتغير الثاني (العين واللام) مما يؤدي إلى تغيير الباب الصرفي لمضارعه، كالأفعال (رَجّ، رنّ)، (لَحّ، لذّ) فإن مضارع الأولين هما (يرُجّ، يرنّ): فيتصرف الفعل (رَجّ) على باب (نصر)، ويتصرف الفعل (رنّ) على باب (ضرب)، كما نجد أن مضارع الفعلين التاليين (يلحّ، يلقّ) مختلف: فيتصرف الفعل (لَحّ) على باب (ضرب)، ويتصرف الفعل (لفّ) على باب (نصر). كما نلاحظ في الأفعال (ذلّ، لذّ)، و(شنّ، نشّ) أن تغيير موقع صوتي المضعف في الفعل الماضي يؤدي إلى اختلاف الباب الصرفي لمضارعه: فالفعل (يذلّ) من باب (ضرب)، والفعل (يلذّ) من باب (فتح)، وكذلك الفعل (يشنّ) من باب (نصر) والفعل (ينشّ) من باب (ضرب).

ومن ثم رأيت أن أرصد جميع الأفعال الثلاثية المضعفة بالقاموس المحيط للفيروز آبادي، متوخية بذلك أن يكتسب البحث طابع الاستقصاء؛ كي يخلص من دراسة المعطيات الشاملة إلى صورة واضحة، يمكن أن تؤدي إلى تحليل دقيق؛ يفضي بنا إلى تلمس الطريق إلى إجابات شافية لتلك التساؤلات، وقد تساعدنا على معرفة بعض القواعد التي تزيح الغموض عن هذا الجانب، وتوضح لنا مدى ارتباط أحياز أصوات المضعف ومخارجها بالباب الصرفي للفعل.

وبدأت بدراسة حيز الحلق وقارنته بحيز الشفتين⁽²⁾، وتوصلت فيه إلى عدد من

(1) وفاء كامل فايد: تراكب الأصوات في الفعل الثلاثي الصحيح - عالم الكتب - القاهرة 1991.

(2) مؤتمر مجمع القاهرة 2009: (أثر تجاور صوتي الفعل الثلاثي المضعف في بابهِ الصرفي: دراسة في حيزي الحلق والشففتين).

النتائج الملموسة. وثبتت بدراسة الأحياز الوسطية فدرست سلوكها الصرفي⁽³⁾، وحددت مدى ارتباط صوتي المضعف ببابه الصرفي في هذه الأحياز. وأتبع ذلك بالدراسة الشاملة لحيز الشفتين⁽⁴⁾، وتوصلت فيها إلى قواعد دقيقة. وأستكمل بدراسة حيز الأصوات الذاقية (ر-ل-ن)؛ حتى تستكمل الدراسة رصد السلوك الصرفي لكل الصوامت في اللغة العربية؛ وهو ما يتيح لنا أن نضيف إلى النتائج السابقة ما يمكن أن يرسم صورة محددة المعالم، تبين أثر صوتي الفعل المضعف في سلوكه على باب صرفي بعينه.

وعرضت- في البحوث السابقة- موقف النحاة وبعض اللغويين القدامى من الفعل الثلاثي المضعف، وربطهم لسلوكه الصرفي بحالته من حيث التعدي واللزوم⁽⁵⁾، وعدم اطمئناني إلى هذا الربط بعد أن اختبرته. كما عرضت **الاتجاه اللغوي المغاير الذي يربط بنية الكلمة بمخارج أصواتها**، كما ورد في مقدمتي معجمي العين والجمهرة : فقد أشار **الخليل** إلى أن العلة في تعذر نطق الأصوات هي قرب مخارجها في الكلمة، وهو ما دعا العرب إلى إهمال بعض الكلمات⁽⁶⁾.

وتبعه **ابن دريد** فرأى أن "الحروف إذا تقاربت مخارجها كانت أثقل على اللسان منها إذا تباعدت"⁽⁷⁾، وذكر أن "أحسن الأبنية عندهم أن يبنوا بامتزاج الحروف المتباعدة"⁽⁸⁾. وإن أراد العرب الجمع بين حرفين من مخرجين متقاربين بدأوا بالأقوى منهما وأخروا الألين. ولم يحدد ابن دريد معايير بعينها لقياس القوة واللين والصعوبة في الحروف.

وتابعهما **ابن جني** فنَبّه إلى نهج العرب في بناء الكلمات، وحدد المستحسن والمستهجى في الأصوات المتجاورة⁽⁹⁾. كما ذكر أن القياس ألا يتألف الحرفان من مخرجين متجاورين، وإن تجشم العرب ذلك بدأوا بالأقوى منهما⁽¹⁰⁾.

ولم يحدد ابن جني معيار قوة الحرف أو ضعفه، ولكن نصه يفهم أن الرء أقوى من اللام والنون⁽¹¹⁾، والشين أقوى من الصاد والسين والزاي⁽¹²⁾، والتاء والطاء أقوى من الدال. وأشار **رضي الدين الاستراباذي** إشارة صريحة إلى أثر مخارج بعض الحروف في حركة عين المضارع من الفعل الصحيح، حين ذكر أن الحروف التي من مخرجي

(3) أثر تجاور صوتي الفعل الثلاثي المضعف في بابيه الصرفي: دراسة في الأحياز الوسطية: مؤتمر مجمع القاهرة 2010.

(4) أثر تجاور صوتي الفعل الثلاثي المضعف في بابيه الصرفي: دراسة في حيز الشفتين: مؤتمر مجمع اللغة العربية بالقاهرة، 2010.

(5) رجعت في ذلك إلى كل من: سيبويه: الكتاب 417/4، المبرد: المقضب 381/1، ابن درستويه: تصحيح الفصح 37، ابن القوطية:

الأفعال 1، ابن جني: الخصائص 379/1، السرقسطي: كتاب الأفعال 57-58، الاستراباذي: شرح الشافية 116/1، 134، ابن

منظور: لسان العرب (ب ت ت)، (ث ر ر) والسيوطي: همع الهوامع 272/3.

(6) الفراهيدي (الخليل بن أحمد): كتاب العين - تحقيق عبد الله درويش - بغداد 1967 : ص 68.

(7) ابن دريد : جمهرة اللغة - دار صادر - بيروت : المقدمة ص 9 .

(8) المرجع السابق : ص 11 .

(9) ابن جني : سر صناعة الإعراب - ت:هنداوى - ط2 - دار القلم - دمشق 1993 : 816/2 ، 65/1 .

(10) المرجع السابق : 814/2 .

(11) المرجع السابق : 818/2 .

(12) نفسه : 817/2 .

الواو أو الياء لا تتغير حركة عين مضارع الفعل الثلاثي الصحيح إلى الكسر أو الضم، كما يفعل حرف الحلق بالضم والكسرة، فيغيرهما إلى الفتحة؛ لتعديل ثقل الحروف الحلقية بخفة الفتحة(13).

كما خلصت دراستي عن تآلف الأصوات وتنافرها في الفعل الثلاثي الصحيح إلى عدد من الأسس التي تحكم تراكب الأصوات في اللغة العربية. ومن هنا حاولت في هذه الدراسة أن أستكمل اختبار هذا الاتجاه اللغوي الذي يربط بنية الكلمة بمخارج أصواتها؛ لكي نتحقق من أثر أصوات الفعل الثلاثي المضعف على الباب الصرفي لمضارعه، سواء أكان الصوت فاء للفعل أم عينا ولاما له، وهو جانب لم يدرس من قبل، فيما أعلم.

أهداف البحث :

يهدف هذا البحث إلى محاولة الإجابة عن التساؤلات التالية :

- 1- هل يؤثر حيز صوتي الفعل الثلاثي المضعف ومخرجهما في ورود الفعل على باب صرفي بعينه؟
- 2- هل كانت أحياز الأصوات ومخارجها صفة حاكمة في اختيار الباب الصرفي للفعل الثلاثي المضعف على لسان العرب القدامى ؟
- 3- على مستوى حيز الأصوات الذلقية (ر ل ن) : هل تكون الصفة الحاكمة لإيثار الفعل بابا صرفيا بعينه، للاتحاد في مخرج أصوات الحيز أم للاختلاف فيه ؟
- 4- هل يمكن تلمس القواعد التي تحكم السلوك الصرفي للفعل الثلاثي المضعف الذي يكون أحد صوتيه من الأصوات الذلقية ؟
- 5- هل يمكن تحديد القواعد التي تربط صوتي الفعل الثلاثي المضعف ببابه الصرفي من خلال برنامج حاسوبي ؟

عينة البحث :

اعتمدت الباحثة (القاموس المحيط) للفيروزابادي لاستقصاء الأفعال الثلاثية الصحيحة المضعفة التي وردت به، لغزارة مادته مع اختصاره(14)، ولحرصه على ضبط حروف كلماته بالشكل، إلى جانب التزامه بتحديد الباب الصرفي لأفعاله.

خطوات البحث :

- استقصت الباحثة الأفعال الثلاثية الصحيحة المضعفة العين واللام في القاموس المحيط، وسجلتها مع تصريفاتها في جدول خاص ارتكز عليه البحث: جدول (1).
- ومن الأفعال المرصودة في الجدول السابق رصدت الباحثة تصرف الفعل الثلاثي المضعف حين تكون الأصوات الذلقية فاءً له، وتتغير أصوات عينه ولامه: جدول (2).
- ومن الجدول (2) سجلت تصرف المضعف الثلاثي حين تكون الأصوات الذلقية عينا ولاما له وتتغير أصوات فائه، ورتبت حروف فاء المضعف ألفائيا؛ حتى يمكن أن يظهر أثر حيز عين المضعف ولامه- دون فائه- في إيثار الفعل بابا صرفيا بعينه: جدول (3).

(13) شرح الشافية : 122/1 .

(14) يحتوى القاموس المحيط على ستين ألف مادة من مواد اللغة.

- ثم أعادت الباحثة ترتيب بيانات الجدول (3)، بحيث رتبت فيه فاء المضعف وفقا لأحيازها؛ وذلك لكي تتبين ما إذا كان إيثار الفعل لباب صرفي بعينه- وفقا لما ظهر بالجدول (3)- يمتد إلى أصوات فاء الفعل التي يضمها حيز واحد، أم أن هذا الاتجاه يختص به صوت بعينه أو أكثر من صوت في الحيز الواحد، دون أن يمتد إلى باقي أصوات الحيز: جدول رقم (4).

- وأخيرا عهدت بالبيانات كلها إلى مبرمج حاسوبي؛ لكي يضع برنامجا يستخرج القواعد والأسس التي يسير عليها الباب الصرفي في الفعل الثلاثي المضعف، الذي يكون أحد صوته من حيز الأصوات الذلقية (ر ل ن)، وهي القواعد التي يحكمها تجاوز صوتي هذا النوع من الأفعال.

ومن خلال البرنامج يمكن تدقيق القواعد التي استخرجتها الباحثة يدويا، سواء أكان ذلك بالتأكيد أو التعديل أو الرفض، أو استخراج قواعد جديدة.

المصطلحات:

قبل عرض نتائج البحث يلزم أن نحدد منظومة المصطلحات المستخدمة فيه، وهي:

المَخْرَج (15) Point of articulation:

هو النقطة التي يلتقي عندها عضوان من أعضاء النطق ليمر هواء الزفير بينهما، ويحدث الصوت.

الحَيِّز (16) range of articulation:

مساحة تشتمل على أكثر من مخرج، وتكون المخارج فيها متقاربة.

الصوت المَهْمُوس (17) voiceless والصوت المَجْهُور (18) voiced.

الإطباق (19) velarisation والانفِتاح (20) Nonvelarisation.

الأَحْيَاؤُ الوَسْطِيَّة:

وتشمل صوتي اللِّهَاءِ وَالْحَنَكِ، والأصوات الشَّجْرِيَّة، والأسْلِيَّة، وَالنَّطْعِيَّة، وَاللَّثَوِيَّة.

وقد استخدمت الباحثة مصطلحات الخليل، واتبعت ترتيبه (21) للأصوات

(15) عرفه ابن يعيش بقوله: " هو المقطع الذي ينتهي الصوت عنده " . شرح المفصل : 124/10.

(16) استخدم الخليل هذا المصطلح بكثرة في كتاب العين ، ص 65/64.

(17) عرفه ابن جنى بأنه : " حرف أضعف الاعتماد في موضعه حتى جرى معه النفس " : سر الصناعة : 60/1 .

(18) عرفه سيبويه بأنه : " الذي يمنع الصوت أن يجرى فيه " : الكتاب 434/4 .

(19) شرح المفصل: 128/10: " والإطباق أن تطبق على مخرج الحرف من اللسان ما حاذاه من الحنك " . أو هو " ارتفاع مؤخر اللسان

إلى أعلى قليلا في اتجاه الطبقة اللينة، وتحركه إلى الخلف قليلا في اتجاه الحائط الخلفي للحلق " .

(20) استخدم سيبويه هذا المصطلح: الكتاب: 436/4. وهناك من يعبر عنه بالترقيق، في مقابل التفخيم .

(21) اختلف سيبويه في ترتيب الصوامت عن الخليل ، وكان ترتيب الحروف عند سيبويه كما يلي :

الهمزة والألف والهاء والعين والحاء والغين والحاء، والقاف والكاف، والجيم والشين والياء، والصاد، واللام والنون والراء، والطاء

والدال والتاء، والزاي والسين والصاد، والطاء والذال والثاء، والفاء والباء والميم والواو: الكتاب 433/4. و سقط مخرج اللام من طبعة

الكتاب، تحقيق (هارون). وقد اتفق ابن جنى مع سيبويه في ترتيبه، واعترض على ترتيب الخليل: سر الصناعة 45/1.

الصامتة، كما ورد في كتاب (العين)، وأضافت إليه الهمزة بترتيب سيبويه، فقسمت الصوامت إلى المجموعات التالية:

1- أصوات الحلق (أ - هـ - ع - ح - غ - خ) :

ويضم حيزها ثلاثة مخارج، أولها: مخرج صوتين من أقصى الحلق، هما الهمزة والهاء⁽²²⁾، والثاني مخرج صوتين من وسط الحلق، هما العين والحاء، والثالث مخرج صوتين من أدنى الحلق، هما الغين والحاء.

2- صوتا اللهاة والحنك الأعلى : (ق - ك) :

يجمعهما حيز واحد⁽²³⁾: القاف اللهوي، ثم الكاف أقصى الحنكي.

3- الأصوات الشجرية⁽²⁴⁾: (ج - ش - ض) .

4- الأصوات الأسلية⁽²⁵⁾: (ص - س - ز) .

5- الأصوات النطعية⁽²⁶⁾: (ط - ت - د) .

6- الأصوات اللثوية⁽²⁷⁾: (ظ - ث - ذ) .

7- الأصوات الذلقية⁽²⁸⁾: (ر - ل - ن) .

(22) الكتاب: 433/4، وفي شرح المفصل: 124/10: " فمن ذلك الحلق وفيه ثلاثة مخارج، فأقصاها من أسفله إلى ما يلي الصدر مخرج الهمزة ولذلك ثقل إخراجها لتباعدتها، ثم الهاء "

(23) الكتاب: 433/4: " ومن أقصى اللسان وما فوقه من الحنك الأعلى مخرج القاف. ومن أسفل من موضع القاف من اللسان قليلا، ومما يليه من الحنك (الأعلى) مخرج الكاف ". وبالمعنى في المقتضب: 328/1 وفي سر الصناعة: 47/1، و شرح المفصل: 124/10، و همع الهوامع: 227/2.

(24) العين: 64، الكتاب 433/4: " ومن وسط اللسان بينه وبين وسط الحنك الأعلى مخرج الجيم والشين والياء ". واتفق معه ابن جني في سر صناعة الإعراب: 46/1. وفي المقتضب قدم مخرج الشين على مخرج الجيم: 328/1، وذكر " أن أقرب الحروف من الياء الجيم " 329/1. وفي شرح المفصل: 124/10 " الجيم والشين والياء ولها حيز واحد، وهو وسط اللسان بينه وبين وسط الحنك، وهي شجرية، والشجر: مفرج الفم، لأن مبدؤها من شجر الفم.. والضاد من حيز الجيم والشين والياء "

(25) العين: 64/1، وتسمى أصوات الصفير. والمقتضب: 329/1، وفي شرح المفصل: 125/10: " الصاد والسين والزاي من حيز واحد، وهو ما بين الثنايا وطرف اللسان، وهي أسلية لأن مبدؤها من أسلة اللسان وهو مستندق طرف اللسان، وهي حروف الصفير "

(26) العين: 64 / 1، شرح المفصل: 125/10: " والطاء والذال والتاء من حيز واحد، هو ما بين طرف اللسان وأصول الثنايا، وهي نطعية لأن مبدؤها من نطح الغار الأعلى، وهو وسطه، يظهر فيه كالتحزيز "

(27) العين: 65 / 1، وتسمى أيضا أصوات ما بين الأسنان. شرح المفصل: 125/10: " والطاء والذال والتاء من حيز واحد، هو ما بين طرف اللسان وأصول الثنايا، وهي لثوية لأن مبدؤها من اللثة "

(28) شاع بين القدماء إطلاق اسم حروف الذلاقة على ستة أصوات هي اللام والراء والنون والفاء والياء والميم: سر صناعة الإعراب: ص 64، وشرح الشافية: 58-257/3. ونسب ابن يعيش إلى سيبويه إطلاق اسم (حروف الذلاقة) على هذه الأصوات التي تجمعها عبارة (مر بنفل) .

ولم تعثر الباحثة في (الكتاب) على ما يشير إلى إطلاق هذه التسمية على تلك الأصوات. ويمكن أن يكون مرجع ذلك إلى أن الخليل، حين تحدث في مقدمة العين عن الحروف الذلقية والشفوية، حددها وذكر سبب التسمية- وهو أن الذلاقة = في

8- الأصوات الشَّفَهِيَّةُ: (ف - ب - م) .

تنبيهات :

1- لما كان الفعل المضعف في صيغة الماضي يختلط فيه كل بابين من الأبواب التالية:

(أ): (نصر) مع (كرم) .

(ب): (فتح) مع (علم) .

(ج): (ضرب) مع (حسب) ، بكسر السين فيهما، بمعنى: ظن.

لاتحادهما في صيغة المضارع؛ ولما كان القاموس المحيط يكتب ماضي الفعل المضعف للغائب- غالبا- فقد آثرت الباحثة أن تكتفي بالأبواب الثلاثة: (نصر) و(ضرب) و(فتح) للمضعف.

على أنها نبهت في الحاشية على صيغة الفعل الذي نص القاموس أو اللسان على تصرفه على باب آخر، أو نسبه إلى ضمير الرفع فظهر بابه الصرفي من صيغة الماضي.

2- يمكن ألا يستدل من ندرة ورود الصوت - كما في حالة الأصوات اللثويَّة (ظ ، ذ ، ث)- على ارتباط الحيز بالصيغة الصرفية. ولكن هذه الندرة تعطي مؤشرا- على الأقل بمقارنة سلوك أصوات الحَيِّزِ نفسها بعضها مع بعض- نلاحظ منه اختلاف السلوك الصرفي لها باختلاف المخارج.

3 - رمزت الباحثة في جداول البحث لكل من الأبواب الصرفية برقم خاص هو :

(1) = نصر . (2) = ضرب . (3) = فتح .

وفيما يلي عرض للجداول التي تناولها البحث بالدراسة

المنطق إنما هي بطرف أسلة اللسان والشفتين، وهما مخرجا هذه الأحرف الستة- ولكنه عاد فقسم تلك الأصوات إلى أصوات ذليقة هي: (ر - ل - ن)، وأصوات شفوية هي: (ب - ف - م) : مقدمة العين ص 57 .

جدول رقم (1)

توزيع الأفعال الثلاثية الصحيحة المضعفة

عين الفعل ولامه

	أ	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	هـ
أ		1 2	1	3 2 1	1 2	1		1		1	1 2	1	3	2 1	2 1	2				2 1		1	2 1	1	2	1
ب			2 1	2 1	1	3	1	1	1	1 3 2	1	1	3	2	2	1	1	2	1		1 2	1	1 2 3		2	3
ت		1					1			2 1								1			1	2 1	2			
ث					1					3 2 1			1			1 2	2					1	1	1		
ج		1	1	1		1	1	1 2	1	1	2	1	1	2	1		1	1		3 2	1		1 2	2 1	2 1	
ح		2 1	1	1	1			1 2	1	3 1 2	1	2 1	1	1	1	1	3			1 2	2 1	1	2 1	1	2	
خ		1	1		1			1	2	1 2	1	1 3 2	1	1		1	1	2		2	2		2 1	1 2	2	
د		2		1	2	1				3 2 1		1	1	1	1		1	1		2 1	2 1	1	2	2 1	1	
ذ		2 1			1	1				1			1						1	2			2	2 1	2	
ر		1	1	2	1		1	1	2		2 1	1	1	1	1					2 1	2	1 2		1 2	2	
ز		1 3	1		1	1	2 1			1	1					2				2 1	1	1 2	2	2 1	1	
س		1			1	2 1	2	1 2		3 1										3 1	1	1	2	1	1	
ش		1 2	2		1 2	2 1 3	1	1	2 1	2 1	2 1	1		2		1 2	1	1 2	1	2	3 1	1	3 1	1	1	
ص		1	2		1	2	2	2 1		1 2									1	1	2	1	1 2	1 3		
ض		3 2 1			2		1	1		1	3							1		1	2	1	2	1	2 3 1	
ط		2 1			1	1				2 1		1	2 1					1		1			3 2 1	1 2	2	
ظ					1					1										1			3		1	
ع		1	2	2	3 2			1		3 2 1	1 3 2	1	1	3	3	1	1			2	2	1 2	1 2	1	1 2	
غ		2	2	3 2				1	2 1	3 1	1	1	1	3	3 1	3 2 1					2		1 2 3	1	3	
ف			1	1	1	2 1	2	1 2	1	2	2		1	2	1		3		1		1	1	2 1		1 3	
ق		1 2	1	1		1		1	1	3 1 2	2 1	1	2 1	2 1	3 1	1 2 3		1		2			2	2 1	1 1	
ك		1	1 2	2	2		2	1	1	1	1	1 3	2	2			1	1 2		1			2	1	1 2	
ل		1	1	1	2	2	1	1	3		1	1	1	1		2	1			1	1	1		1	1	
م			1	1	1	3 2 1		1		3 1	3 1	1 3	1	1	1	1	1	1			1	1	3 2 1		1 3	
ن		2	1	2 1	2	1 2	2	2	1		2	2 1	2	2 1	2	2				1	2			1 2		
هـ		1 2	2	1	1			2 3 1	1	3 2 1	1	2	2 1	1	1					2	1	1	2	1 2	2 3	

نصر ينصر (1)
ضرب يضرب (2)
فتح يفتح (3)

كتبت الأفعال حسب ترتيب ورودها بالقاموس. والترتيب مهم؛ حيث إنه قدم المشهور الفصيح أولاً.

جدول رقم (2)
تقسيم المضعف الثلاثي الصحيح مع الأصوات الذلقية (ر ل ن) فاء له

حيز أصوات عين الفعل ولامه															فء المضعف											
الشفتان			حروف الذلاقة			الثثة			نطع الغار			الأسلة				شجر الفم			حنك ولهة		الحلق					(تلقى)
م	ب	ف	ن	ل	ر	ذ	ظ	ث	د	ط	ت	ز	ص	س	ض	ج	ش	ك	ق	خ	غ	ح	ع	هـ	أ	
1	1	1	-	-	-	-	-	-	1	1	-	1	1	1	1	1	1	1	1	1	-	-	-	-	-	راء
2		2	2			2	2					2						2	2							
1	1	1	-	-	-		1	1	1	1	2	1	1	1	-	-		1	1	1	-		-	1	-	لام
						3										2						2				
1		1	-	-	-	-	1	-		1			1	1				-			-		-	-	-	نون
2	2						2		2	2		2	2	2	2	2	2	2	2	2	2	2				

جدول رقم (3)
تقسيم المضعف الثلاثي الصحيح مع الأصوات الذلّقية عينا ولاما

الأصوات الذلّقية عينا ولاما للمضعف (مع صفاتها)			فاء المضعف (ألفبائيا)
النون	اللام	الراء	
متوسط/ مجهور/ خيشومي	متوسط/ مجهور/ جانبي	متوسط/ مجهور/ مكرر	
	1	1	الهمزة
2	2		
	1	1	الباء
2	2	2	
	3	3	
	1	1	التاء
-	2	2	
-	1	1	الثاء
		2	
		3	
1	1	1	الجيم
2	2		
	1	1	الحاء
2	2	2	
	3	3	
1	1	1	الخاء
2	2	2	
1		1	الذال
	2	2	
		3 نادر	
		1	الذال
2	2		
		3 شاذ	
	-	-	الراء
2			
1		1	الزاي
	2		
1		1	السين
	2	3	
1	1	1	الشين
		2	
	3		

تابع جدول رقم (3)

تقسيم المضعف الثلاثي الصحيح مع الأصوات الذلقية عينا ولاما

الأصوات الذلقية عينا ولاما للمضعف (مع صفاتها)			فء المضعف (ألفبائيا)
النون	اللام	الراء	
متوسط/ مجهور/ خيشومي	متوسط/ مجهور/ جانبي	متوسط/ مجهور/ مكرر	
-	1 2	1 2	الصاد
2 3	2	1	الضاد
2	1 2 3	1 2	الطاء
1	3	1	الظاء
1 2	1 2	1 2 3	العين
3	1 2 3	1 3	الغين
1	1 2	2	الفاء
1	2	1 2 3	القاف
1	2	1	الكاف
-	-	-	اللام
1	1 3	1 3	الميم
-	-	-	النون
2	2	1 2 3	الهاء

جدول رقم (4)

تقسيم المضعف الثلاثي الصحيح مع الأصوات الذلقية عينا ولاما

الأصوات الذلقية عينا ولاما للمضعف (متوسطة)			فاء المضعف	حيز الفاء
النون	اللام	الراء		
مجهور/ خيشومي	مجهور/ جانبي	مجهور/ مكرر		
2	1 2	1	الهمزة	الحلق: (أقصى الحلق)
2	2	1 2 3	الهاء	
1 2	1 2	1 2 3	العين	(وسط الحلق)
2	1 2 3	1 2 3	الحاء	
3	1 2 3	1 3	الغين	(أدنى الحلق)
1 2	1 2	1 2	الخاء	
1	2	1 2 3	القاف	اللهاة والحنك الأعلى
1	2	1	الكاف	
1 2	1 2	1	الجيم	شجر الفم
1	1 3	1 2	الشين	
2 3	2	1	الضاد	
-	1 2	1 2	الصاد	الأسلة

تابع جدول رقم (4)

تقسيم المضعف الثلاثي الصحيح مع الأصوات الذلقية عينا ولاما

الأصوات الذلقية عينا ولاما للمضعف (متوسطة)			فاء المضعف	حيز الفاء
النون	اللام	الراء		
مجهور/ خيشومي	مجهور/ جانبي	مجهور/ مكرر		
1	2	1 3	السين	الأسلة
1	2	1	الزاي	
2	1 2 3	1 2	الطاء	نطع الغار
-	1 2	1 2	التاء	
1	2	1 2 3 نادر	الذال	
1	3	1	الظاء	الثلثة
-	1	1 2 3	الثاء	
2	2	1 3 شاذ	الذال	
2	-	-	الراء	ذولق اللسان
-	-	-	اللام	
-	-	-	النون	
1	1 2	2	الفاء	الشفقتان
2	1 2 3	1 2 3	الباء	
1	1 3	1 3	الميم	

نتائج البحث:

من الجدول (2) نلاحظ السلوك التالي للأصوات الذلقية (المجهورة) في موقع الفاء:
I - فاء المضعف من الأصوات الذلقية (ر- ل- ن):

أولاً: مع الأصوات الحلقية عينا ولاما للمضعف:

1. تتناظر الذلقيات (المجهورة) (ر-ل-ن)- فاء- مع الحلقيات المجهورة (أ-ع-غ).
2. لا تقع الأصوات الذلقية (ر-ل-ن) فاء مع (الهمزة) و(الهاء)⁽²⁹⁾ أقصى الحلقين.
3. لا تقع الأصوات الذلقية المجهورة (ر-ل-ن) فاء مع (ع) وسط الحلقى المجهور.
4. لا يقع (الراء) المجهور المكرر فاء مع (الحاء) وسط الحلقى المهموس.
5. لا يتصرف (اللام) المجهور الجانبي- فاء- مع (الحاء) وسط الحلقى المهموس إلا على باب (ضرب)، في الفعل (لحّ).
6. يتصرف (النون) المجهور الخيشومي- فاء- مع (الحاء) وسط الحلقى المهموس على بابي (نصر) و(ضرب) في الفعل (نحّ)⁽³⁰⁾.
7. لا تقع الأصوات الذلقية (ر-ل-ن) فاء مع (غ) أدنى الحلقى المجهور المستعلي.
8. يتصرف الذلقيان غير الخيشوميين (ر-ل)- فاء- مع (الخاء) أدنى الحلقى المهموس المستعلي على باب (نصر) منفردا، في الفعلين (رَحّ، لَحّ).
9. لا يتصرف (النون) الذلقى الخيشومي- فاء- مع (الخاء) الحلقى المهموس المستعلي إلا على باب (ضرب)، في الفعل (نحّ).
10. يتصرف (النون) الذلقى الخيشومي- فاء- مع المهموسين من وسط الحلق وأدناه (ح-خ) على باب (ضرب)، في الفعلين (نحّ⁽³¹⁾، نحّ).
11. لا يتصرف (اللام) المجهور الجانبي- فاء- مع (ح) وسط الحلقى المهموس المستقل إلا على (ضرب)، في الفعل (لحّ)، ولا يتصرف (اللام) مع (الخاء) أدنى الحلقى المهموس المستعلي إلا على (نصر)، في الفعل (لحّ).

ثانياً: مع صوتي اللهاة والحنك الأعلى عينا ولاما للمضعف:

1. لا يتصرف الذلقيان المجهوران (ر-ن)- فاء- مع (القاف) اللهوي المجهور إلا على باب (ضرب)، في الفعلين (رقّ، نقّ).
2. لا يتصرف (اللام) الذلقى الجانبي- فاء- مع (ق-ك) إلا على (نصر)، في الفعلين (لقّ، لكّ).
3. لا يقع (النون) الذلقى الخيشومي المجهور فاء مع (الكاف) الحنكي المهموس.
4. يتصرف (الراء) الذلقى المكرر المجهور- فاء- مع (الكاف) الحنكي المهموس على بابي (نصر) و(ضرب) في الفعل (ركّ)⁽³²⁾.

ثالثاً: مع الأصوات الشجرية عينا ولاما للمضعف:

1. لا يتصرف (الراء) الذلقى المكرر المجهور- فاء- مع الأصوات الشجرية (ج-ش-ض) إلا على (نصر)، في الأفعال (رَجّ-رَشّ-رَضّ).

(29) لم يستثن من ذلك سوى الفعل (لّة) الذي أورده القاموس المحيط متصرفا على باب (نصر)، وأورده تاج العروس متصرفا على باب

(ضرب)، ولم يرد في لسان العرب.

(30) يختلف معنى الفعل (نحّ) على البابين.

(31) يتصرف الفعل (نحّ) على باب (نصر) أيضا في القاموس المحيط، ولم يرد في لسان العرب إلا على باب (ضرب).

(32) يختلف معنى (ركّ) على البابين.

2. لا يتصرف (النون) الذلقي الخيشومي المجهور- فاء- مع الأصوات الشجرية (ج- ش- ض) إلا على (ضرب)، في الأفعال (نَجّ- نشّ- نضّ).
3. لا يقع (اللام) الجانبي المجهور فاء للمضعف مع (ش)⁽³³⁾ و(ض)⁽³⁴⁾ الشجريين.
4. لا يتصرف (اللام) الذلقي الجانبي المجهور- فاء- مع (الجيم) الشجري المزجي المجهور المنفتح إلا على (ضرب) في الفعل (لجّ).

رابعاً: مع الأصوات الأصلية عينا ولأما للمضعف:

1. لا يتصرف (اللام) الذلقي الجانبي المجهور- فاء- مع الأصوات الأصلية (ص- س- ز) إلا على باب (نصر)، في الأفعال (لصّ- لسّ- لزّ).
2. لا يتصرف (الراء) الذلقي المكرر المجهور- فاء- مع الأسليين المهموسين (ص- س) إلا على باب (نصر)، في الفعلين (رصّ، رسّ).
3. يتصرف (الراء) الذلقي المكرر المجهور- فاء- مع الأسلي المجهور (ز) على بابي (نصر) و(ضرب) في الفعل (رزّ⁽³⁵⁾).
4. لا يتصرف (النون) الخيشومي المجهور- فاء- مع (الزاي) الأسلي المجهور إلا على (ضرب) في الفعل (نزّ).
5. يتصرف (النون) الخيشومي المجهور- فاء- مع الأسليين المهموسين (ص- س) على بابي (نصر، ضرب) في الفعلين (نصّ⁽³⁶⁾، نسّ⁽³⁷⁾).
6. لا يتصرف الذلقيان المجهوران غير الخيشوميين (ر- ل) مع الأسليين المهموسين (ص- س) إلا على (نصر)، في الأفعال (رصّ، رسّ)، (لصّ، لسّ).

خامساً: مع الأصوات النطعية عينا ولأما للمضعف:

1. لا يقع (الراء) الذلقي المكرر المجهور فاء مع (الطاء) النطعي المجهور المطبق.
2. لا يتصرف (ل- ن) المجهوران- فاء- مع (ط) المجهور المطبق إلا على (ضرب)، في الفعلين (لطّ، نطّ).
3. لا يتصرف (ن) الخيشومي المجهور- فاء- مع النطعيين المجهورين (ط- د) إلا على (ضرب) في الفعلين (نطّ، ندّ).
4. لا يتصرف (ن) الخيشومي- فاء- مع النطعي المهموس (ت) إلا على (نصر)، في الفعل (نتّ).

(33) نقل الخفاجي عن المحكم أن ليس في كلام العرب شين بعد لام في كلمة عربية: انظر شفاء الغليل - ص8. وقد تأكدت هذه العبارة حين أظهر استقصاء الأفعال الثلاثية الصحيحة في القاموس المحيط أن اللام لا يقع فاء لفعل عينه شين، أو عينا لفعل لاهه شين، أو فاء لفعل لاهه شين. انظر تراكم الأصوات في الفعل الثلاثي الصحيح: ص109.

(34) بسبب قرب مخرجيهما: ذكر سيبويه أن " (الضاد) استطالت لرخوتها حتى اتصلت بمخرج (اللام) ": الكتاب ج4- ص457 وأيضا في شرح شافية ابن الحاجب- ج 3 ص279.

(35) يختلف معنى (رزّ) على البابين، وقد أورده القاموس متصرفا على البابين، في حين لم يرد باللسان إلا على باب (نصر).

(36) يختلف معنى (نصّ) على البابين.

(37) يختلف معنى (نسّ) على البابين.

5. لا يتصرف الذلقيان غير الخيشوميين (ر- ل) مع النطعيين المنفتحين (ت- د) إلا على (نصر) في الأفعال (رت، رد)، (لت، لد).
6. لا يتصرف (اللام) الجانبي مع (الطاء) النطعي المطبق إلا على (ضرب)، في الفعل (لظ)، ولا يتصرف مع النطعيين المنفتحين (ت - د) إلا على (نصر)، في الفعلين (لت، لد).

سادسا: مع الأصوات اللثوية عينا ولاما للمضعف:

1. لا يقع (الراء) المكرر المجهور فاء مع (الطاء) اللثوي المجهور المطبق.
2. لا يتصرف (الراء) - فاء- مع اللثويين المنفتحين (ث- ذ) إلا على (ضرب) منفردا، في الفعلين (رت، رد).
3. لا يقع (النون) الذلقي الخيشومي المجهور فاء مع اللثويين المجهورين (ظ- ذ).
4. يتصرف (ن) الخيشومي- فاء- مع (ث) اللثوي المهموس على البابين (نصر) و(ضرب)، في الفعل (نت⁽³⁸⁾).
5. لا يتصرف (اللام) الجانبي المجهور- فاء- مع اللثوي المجهور المطبق (ظ)، وكذلك مع اللثوي المهموس المنفتح (ث) إلا على باب (نصر) في الفعلين (لظ- لت).
6. لا يتصرف (اللام) الجانبي المجهور- فاء- مع (ذ) اللثوي المجهور المنفتح إلا على (فتح)، في الفعل (لد).

سابعا: مع الأصوات الذلقية عينا ولاما للمضعف:

1. لا يقع الذلقيان (ل- ن) فاء للمضعف مع الأصوات الذلقية الثلاثة: (ر- ل- ن).
2. لا يقع (الراء) الذلقي المكرر فاء للمضعف مع الذلقيين (الراء) و(اللام).
3. يتصرف (الراء) المكرر- فاء- مع (ن) الخيشومي على(ضرب)، في الفعل (رن).

ثامنا: مع الأصوات الشفهية عينا ولاما للمضعف:

1. لا يتصرف (اللام) الجانبي- فاء- مع الأصوات الشفهية (ف- ب- م) إلا على (نصر)، في الأفعال (لف، لب، لم).
2. يتصرف (الراء) الذلقي المكرر- فاء- مع الأصوات الشفهية (ف- ب- م) على (نصر)، في الأفعال (رف⁽³⁹⁾، رب، رم⁽⁴⁰⁾).
3. لا يتصرف (ن) الخيشومي- فاء- مع (ف) الاحتكاكي المهموس إلا على (نصر)، في الفعل (نف).
4. لا يتصرف (النون) الخيشومي المتوسط- فاء- مع (الباء) الانفجاري المجهور إلا على (ضرب)، في الفعل (نّب).
5. لا يتصرف الذلقيان المتوسطان غير الخيشوميين (ر- ل)- فاء- مع (ب) الانفجاري إلا على (نصر)، في الفعلين (رب، لب).
6. يتصرف الذلقيان: المكرر(ر) والخيشومي (ن)- فاء- مع (الميم) الشفهي الخيشومي على بابي (نصر) و(ضرب)، في الفعلين (رم، نم⁽⁴¹⁾).

(38) يختلف معنى (نتّ) على البابين.

(39) يتصرف الفعل (رف) أيضا على باب (ضرب)، وقد اتفق البابين في معان واختلفا في معان أخر.

(40) اتفق بابا (نصر و ضرب) في معنى من معاني الفعل (رم)، واختلفا في باقي المعاني.

ومن الجدولين (3)،(4) نلاحظ الاتجاهات التالية للأصوات الذلقية المجهورة، عينا ولاما للمضعف:

I - عين المضعف ولامه من الأصوات الذلقية (ر- ل - ن):

أولاً: مع الأصوات الحلقية فاءً للمضعف:

1. لا يمتنع تصرف كل الأصوات الحلقية- فاءً- مع الأصوات الذلقية الثلاثة (ر- ل- ن).
2. لا يقع (الهمزة) أقصى الحلقي المجهور- فاءً- على باب (نصر) منفرداً إلا مع (الراء) المكرر، في الفعل (أرّ).
3. لا يتصرف (أ) أقصى الحلقي- فاءً- مع (ن) الخيشومي إلا على (ضرب) في الفعل (أنّ).
4. لا يتصرف (الهاء) أقصى الحلقي المهموس- فاءً- مع (ل) الجانبي، و(ن) الخيشومي إلا على (ضرب)، في الفعلين (هلّ، هنّ).
5. يتصرف (العين) وسط الحلقي المجهور- فاءً- مع (ر- ل- ن) على بابي (نصر) و(ضرب)، في الأفعال (عرّ (42)، علّ (43)، عنّ (44)).
6. يتصرف (الحاء) المهموس- فاءً- مع الذلقيين غير الخيشوميين (ر- ل) على الأبواب (نصر) و(ضرب) و(فتح)، في الفعلين (حرّ (45)، حلّ (46)).
7. لا يتصرف (الحاء)- فاءً- مع (النون) الخيشومي إلا على (ضرب) في الفعل (حنّ).
8. يتصرف (الغين) المجهور المستعلي- فاءً- مع الذلقيين غير الخيشوميين (ر- ل) على (نصر) و(فتح)، في الفعلين (غرّ (47)، غلّ (48)).
9. لا يتصرف (الغين) المستعلي- فاءً- مع (ن) الخيشومي إلا على (فتح) في الفعل (غنّ).
10. يتصرف (الخاء) أدنى الحلقي المهموس المستعلي- فاءً- مع الأصوات الذلقية المجهورة (ر- ل- ن) على بابي (نصر و(ضرب))، في الأفعال (خرّ (49)، خلّ (50)، خنّ (51)).
11. تتصرف أصوات الحلق (ه- ع - ح -) فاءً- مع (الراء) الذلقي المكرر على الأبواب (نصر) و(ضرب) و(فتح)، في الأفعال (هرّ (52)- عرّ (53)- حرّ (54)).

(41) اتفق بابا (نصر و(ضرب)) في معنى من معاني الفعل (نمّ)، واختلفا في غيره، كما انفرد باب (نصر) بمعنى.

(42) يتصرف (عرّ) أيضاً على(فتح)، وقد اتفق (نصر) و(ضرب) في أحد معانيه، واختلفت الأبواب الثلاثة في باقي المعاني.

(43) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (علّ)، واختلفا في باقي المعاني.

(44) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (عنّ)، واختلفا في باقي المعاني.

(45) ومعنى الفعل (حرّ) واحد على الأبواب الثلاثة.

(46) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (حلّ)، واختلفت الأبواب الثلاثة في باقي المعاني.

(47) يختلف معنى الفعل (غرّ) على البابين.

(48) يتصرف الفعل (غلّ) على باب (ضرب) أيضاً، ومعناه مختلف على الأبواب الثلاثة.

(49) ومعنى الفعل (خرّ) واحد على البابين.

(50) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (خلّ)، واختلفا في باقي المعاني.

(51) يختلف معنى الفعل (خنّ) على البابين.

(52) يختلف معنى الفعل (هرّ) على الأبواب الثلاثة.

(53) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (عرّ)، واختلفت الأبواب الثلاثة في باقي المعاني.

(54) ومعنى الفعل (حرّ) واحد على الأبواب الثلاثة.

12. لا تتصرف أصوات الحلق (الهمزة والهاء والحاء)- فاء- مع (ن) الخيشومي إلا على (ضرب)، في الأفعال (أن- هن- حن).
13. يتصرف الحلقيان المجهوران المستقلان (الهمزة والعين)- فاء- مع (اللام) الجانبي على بابي (نصر) و(ضرب) في الفعلين (أل⁽⁵⁵⁾، عل⁽⁵⁶⁾).

ثانيا: مع صوتي اللهاة والحنك الأعلى فاءً للمضعف:

1. لا يتصرف (ق- ك)- فاء- مع (ل) الجانبي إلا على (ضرب)، في الفعلين (قل، كل).
2. لا يتصرف (ق- ك)- فاء- مع (ن) الخيشومي إلا على (نصر)، في الفعلين (قن، كن).
3. يتصرف (ق) اللهوي المجهور المستعلي- فاء- مع (الراء) المكرر على الأبواب (نصر و(ضرب وفتح)، في الفعل (قر⁽⁵⁷⁾).
4. لا يتصرف (الكاف) الحنكي المهموس- فاء- مع (ر) المكرر و(ن) الخيشومي إلا على (نصر)، في الفعلين (كر، كن).

ثالثا: مع الأصوات الشجرية فاءً للمضعف:

1. لا يتصرف (ج) المجهور- فاء- مع (الراء) المكرر إلا على (نصر)، في الفعل (جر).
2. لا يتصرف (ض) المجهور المطبق- فاء- مع (ر) إلا على (نصر)، في الفعل (ضر).
3. لا يتصرف (ض) المطبق- فاء- مع (ل) الجانبي إلا على (ضرب) في الفعل (ضل).
4. لا يتصرف (الشين) الاحتكاكي المهموس- فاء- مع (النون) الذلقي الخيشومي إلا على (نصر) في الفعل (شن).
5. يتصرف (الجيم) المزجي المجهور- فاء- مع (ل- ن) على بابي (نصر و(ضرب) في الفعلين (جل⁽⁵⁸⁾، جن⁽⁵⁹⁾).
6. يتصرف (ش) الاحتكاكي المهموس- فاء- مع (ر) المكرر على (نصر- ضرب) في الفعل (شر⁽⁶⁰⁾).
7. يتصرف (ش) الاحتكاكي المهموس- فاء- مع (ل) الجانبي على (نصر وفتح) في الفعل (شل⁽⁶¹⁾).
8. يتصرف (ض) المجهور المطبق- فاء- مع (ن) الخيشومي على بابي (ضرب وفتح) في الفعل (ضن⁽⁶²⁾).
9. تتصرف (ج- ش- ض)- فاء- مع (ر) المكرر على (نصر) في الأفعال (جر، شر⁽⁶³⁾، ضر).

(55) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (أل)، واختلفا في باقي المعاني.

(56) اتفق بابا (نصر) و(ضرب) في معنى من معاني الفعل (عل)، واختلفا في غيره.

(57) يختلف معنى الفعل (قر) على الأبواب الثلاثة.

(58) يختلف معنى الفعل (جل) على البابين.

(59) يختلف معنى الفعل (جن) على البابين.

(60) اتفق بابا (نصر) و(ضرب) في بعض معاني الفعل (شر)، واختلفا في باقي المعاني.

(61) يختلف معنى الفعل (شل) على البابين.

(62) ومعنى الفعل (ضن) واحد على البابين.

رابعاً: مع الأصوات الأслية فاءً للمضعف:

1. لا يقع (ص) الاحتكاكي المهموس المطبق فاء مع (ن) المتوسط المجهور الخيشومي.
2. يتصرف الأسلان الاحتكاكيان المنفتحان (س- ز)- فاء- مع (ن) الخيشومي على (نصر) منفرداً، في الفعلين (سنّ، زنّ).
3. لا يتصرف (ز) الاحتكاكي المجهور- فاء- مع المكرر (ر) والخيشومي (ن) إلا على (نصر) في الفعلين (زرّ، زنّ).
4. لا يتصرف (ز) المجهور- فاء- مع (ل) الجانبي إلا على (ضرب) في الفعل (زلّ).
5. يتصرف (الصاد) المهموس المطبق- فاء- مع الذلقين غير الخيشوميين (ر- ل) على بابي (نصر وضرب)، في الفعلين (صرّ⁽⁶⁴⁾، صلّ⁽⁶⁵⁾).
6. لا يتصرف (س) المهموس المنفتح- فاء- مع (اللام) الجانبي إلا على (ضرب) في الفعل (سلّ).
7. يتصرف (س) المهموس- فاء- مع (ر) المكرر على (نصر وفتح) في الفعل (سرّ⁽⁶⁶⁾).

خامساً: مع الأصوات النطعية فاءً للمضعف:

1. لا يقع (ت) الانفجاري المهموس المنفتح فاء مع (ن) المتوسط الخيشومي المجهور.
2. يتصرف (التاء) الانفجاري المهموس- فاء- مع الذلقين المتوسطين غير الخيشوميين (ر- ل) على بابي (نصر) و(ضرب)، في الفعلين (ترّ⁽⁶⁷⁾، تلّ⁽⁶⁸⁾).
3. لا يتصرف (ط) الانفجاري المجهور المطبق- فاء- مع (ن) المتوسط الخيشومي المجهور إلا على (ضرب) في الفعل (طنّ).
4. يتصرف (الطاء) الانفجاري المجهور المطبق- فاء- مع (الراء) المتوسط المكرر المجهور على بابي (نصر) و(ضرب) في الفعل (طرّ⁽⁶⁹⁾).
5. يتصرف (الطاء) الانفجاري المجهور المطبق- فاء- مع (اللام) المتوسط الجانبي المجهور على الأبواب (نصر وضرب وفتح) في الفعل (طلّ⁽⁷⁰⁾).
6. لا يتصرف (الدال) الانفجاري المجهور المنفتح- فاء- مع (النون) المتوسط الخيشومي المجهور إلا على (نصر) في الفعل (دنّ).
7. لا يتصرف (الدال) الانفجاري المجهور- فاء- مع (اللام) المتوسط الجانبي المجهور إلا على (ضرب) في الفعل (دلّ).
8. يتصرف (الدال)- فاء- مع (ر) المتوسط المكرر المجهور على الأبواب الثلاثة، في الفعل (درّ⁽⁷¹⁾).

(63) سبق الإشارة إلى أن الفعل (شرّ) يتصرف أيضاً على (ضرب)، وإلى اختلاف معانيه على البابين.

(64) يختلف معنى الفعل (صرّ) على البابين.

(65) يختلف معنى الفعل (صلّ) على البابين.

(66) يختلف معنى الفعل (سرّ) على البابين.

(67) يتفق معنى الفعل (ترّ) على البابين.

(68) يتفق معنى الفعل (تلّ) على البابين.

(69) يتفق معنى الفعل (طرّ) على البابين.

(70) يختلف معنى الفعل (طلّ) على الأبواب الثلاثة.

9. تتصرف الأصوات النطعية (ط ت د- فاء- مع (ر) على البابين (نصر) و(ضرب) في الأفعال (طرّ، ترّ، ذرّ⁽⁷²⁾).

سادسا: مع الأصوات اللثوية فاءً للمضعف:

1. لا يتصرف (ظ) الاحتكاكي المجهور المطبق- فاءً- مع (ن) المتوسط الخيشومي المجهور إلا على (نصر) في الفعل (ظنّ).
2. لا يتصرف (ذ) الاحتكاكي المجهور المنفتح- فاءً- مع (ن) المتوسط الخيشومي المجهور إلا على(ضرب) في الفعل (ذنّ).
3. لا يقع (الثاء) الاحتكاكي المهموس فاء مع (النون) المتوسط الخيشومي المجهور.
4. لا يتصرف (ث) الاحتكاكي المهموس المنفتح- فاء- مع (ل) المتوسط الجانبي المجهور إلا على (نصر) في الفعل (ثلّ).
5. لا يتصرف (ذال) الاحتكاكي المجهور المنفتح- فاءً- مع (ل) إلا على (ضرب) في الفعل (ذلّ).
6. لا يتصرف(ظ) الاحتكاكي المجهور المطبق- فاءً- مع (ل) إلا على (فتح) في: (ظلّ).
7. لا يتصرف (الظاء) الاحتكاكي المجهور المطبق- فاءً- مع (ر) المتوسط المكرر المجهور إلا على (نصر) في الفعل (ظرّ).
8. يتصرف (ذال) الاحتكاكي المجهور المنفتح- فاءً- مع (ر) المتوسط المكرر المجهور على (نصر) و(فتح)⁽⁷³⁾، في الفعل (ذرّ⁽⁷⁴⁾).
9. يتصرف (الثاء) الاحتكاكي المهموس المنفتح- فاء- مع (الراء) المتوسط المكرر المجهور على الأبواب الثلاثة (نصر) و(ضرب) و(فتح)، في الفعل (ثرّ⁽⁷⁵⁾).
10. لا يتصرف (ذال) الاحتكاكي المجهور المنفتح- فاءً- مع (ل) المتوسط الجانبي المجهور، و(ن) المتوسط الخيشومي المجهور إلا على (ضرب)، في الفعلين (ذلّ، ذنّ).

سابعا: مع الأصوات الذلقية فاءً للمضعف:

1. لا يقع أحد الأصوات الذلقية (ر- ل- ن) فاءً مع (الراء واللام) غير الخيشوميين.
2. لا يقع (اللام والنون) الذلقيان فاءً مع الأصوات الذلقية الثلاثة (ر- ل- ن).
3. يتصرف (الراء) المتوسط المكرر المجهور- فاءً- مع (النون) المتوسط الخيشومي المجهور على (ضرب)، في الفعل (رنّ)⁽⁷⁶⁾.

ثامنا: مع الأصوات الشفهية فاءً للمضعف:

1. لا يتصرف (الميم) المتوسط المجهور الخيشومي- فاءً- مع (النون) الذلقي الخيشومي المتوسط المجهور إلا على باب (نصر)، في الفعل (منّ).

(71) اتفق بابا (نصر) و(ضرب) في بعض معاني الفعل (ذرّ)، واختلفت الأبواب الثلاثة في باقي المعاني.

(72) أشار القاموس إلى أن باب (فتح) نادر في الفعل (ذرّ).

(73) أشار القاموس إلى أن باب (فتح) شاذ في الفعل (ذرّ).

(74) يختلف معنى الفعل (ذرّ) على البابين.

(75) ومعنى الفعل (ثرّ) واحد على الأبواب الثلاثة.

(76) هي الحالة الوحيدة التي لا تتنافر فيها أصوات الحيز الواحد.

2. لا يتصرف (الباء) الانفجاري المجهور- فاء- مع (النون) الذلقي الخيشومي المتوسط المجهور إلا على باب (ضرب)، في الفعل (بَنّ).
3. لا يتصرف (الفاء) الاحتكاكي المهموس- فاء- مع (النون) الذلقي الخيشومي المتوسط المجهور إلا على باب (نصر)، في الفعل (فَنّ).
4. لا يتصرف (الفاء) الاحتكاكي المهموس- فاء- مع (الراء) المتوسط المكرر المجهور إلا على باب (ضرب)، في الفعل (فَرّ).
5. يتصرف (الفاء) الاحتكاكي المهموس- فاء- مع (اللام) المتوسط المجهور الجانبي على بابي (نصر) و(ضرب)، في الفعل (فَلّ⁽⁷⁷⁾).
6. يتصرف (الميم) المتوسط المجهور الخيشومي- فاء- مع الذلقين المتوسطين غير الخيشوميين (ر- ل) على بابي (نصر) و(فتح)، في الفعلين (مَرّ⁽⁷⁸⁾، مَلّ⁽⁷⁹⁾).
7. يتصرف (الباء) الشفهي الانفجاري المجهور- فاء- مع المتوسطين المجهورين غير الخيشوميين: (ر- ل) على الأبواب (نصر- ضرب- فتح) في الفعلين (بَرّ⁽⁸⁰⁾، بَلّ⁽⁸¹⁾).

الخلاصة

من استقراء النتائج السابقة يمكن أن نجل الاتجاهات العامة لارتباط الأصوات الذلقية بأبوابها الصرفية على النحو التالي:

أولاً : ارتباط مخرج فاء المضعف مع مخرج عينه ولامه (مع ملاحظة أثر صفاتهما):

I - فاء المضعف من الأصوات الذلقية المتوسطة المجهورة (ر- ل- ن):

1. لا يقع (الراء) المكرر المجهور فاءً للمضعف مع (الطاء) النطعي المجهور المطبق.
2. لا يقع (الراء) المكرر المجهور فاءً للمضعف مع (الظاء) اللثوي المجهور المطبق.
3. لا يقع (النون) الخيشومي المجهور فاءً مع (الكاف) الحنكي المهموس. ولا يتصرف (النون)- فاء- مع (القاف) اللهوي المجهور إلا على (ضرب).
4. لا يتصرف (الراء) المكرر المجهور- فاء- مع (القاف) اللهوي المجهور إلا على (ضرب). ولا يتصرف (اللام) الجانبي المجهور- فاء- مع (القاف) إلا على (نصر).
5. لا يتصرف (ل) الجانبي- فاء- مع (ح) الحلقي المهموس المستقل إلا على (ضرب). ولا يتصرف (ل) مع (خ) الحلقي المهموس المستعلي إلا على (نصر).
6. لا يتصرف (ن) الخيشومي- فاء- مع (خ) الحلقي المهموس المستعلي إلا على (ضرب).
7. لا يقع (اللام) الجانبي فاءً للمضعف مع (الضاد) الشجري المجهور المطبق⁽⁸²⁾.
8. لا يتصرف (اللام) الجانبي- فاء- مع (الجيم) الشجري المجهور إلا على (ضرب).

(77) يختلف معنى الفعل (فَلّ) على البابين.

(78) يختلف معنى الفعل (مَرّ) على البابين.

(79) يختلف معنى الفعل (مَلّ) على البابين.

(80) يتفق البابين (ضرب) و(فتح) في المعنى، ويختلف معهما معنى الفعل على باب (نصر).

(81) يختلف معنى الفعل (بَلّ) على الأبواب الثلاثة.

(82) بسبب قرب مخرجيهما : ذكر سيويوه أن "(الضاد) استطالت لرخوتها حتى اتصلت بمخرج (اللام)": الكتاب ج4- ص457.

9. لا يتصرف (النون) الخيشومي- فاء- مع (الذال) إلا على باب (ضرب).
10. لا يتصرف (اللام) الجانبي- فاء- مع (ذ) اللثوي المجهور المنفتح إلا على (فتح).
11. لا يتصرف (اللام) الجانبي- فاء- مع (الزاي) الأسلي المجهور إلا على (نصر).
12. لا يتصرف (النون) الخيشومي- فاء- مع (الزاي) الأسلي المجهور إلا على (ضرب).
13. لا يتصرف (النون)- فاء- مع (الباء) الانفجاري المجهور إلا على (ضرب).
14. لا يتصرف (الراء) المكرر- فاء- مع (النون) الذلقي الخيشومي إلا على (ضرب).

II - عين المضعف ولامه من الأصوات الذلقة المتوسطة المجهورة (ر- ل- ن):

1. لا يتصرف (الهمزة) أقصى الحلقي- فاء- مع (الراء) المكرر إلا على باب (نصر).
2. ولا يتصرف (الهمزة)- فاء- مع (النون) الخيشومي إلا على باب (ضرب).
3. لا يتصرف (ح) وسط الحلقي المهموس- فاء- مع (ن) الخيشومي إلا على (ضرب).
4. لا يتصرف (غ) الحلقي المجهور المستعلي- فاء- مع (ن) الخيشومي إلا على (فتح).
5. لا يتصرف (الكاف) المهموس- فاء- مع (اللام) الجانبي إلا على باب (ضرب).
6. لا يتصرف (ش) الشجري المهموس- فاء- مع (ن) الخيشومي المجهور إلا على (نصر).
7. لا يتصرف (ض) الشجري المجهور المطبق- فاء- مع (ر) المكرر إلا على (نصر).
8. لا يتصرف (الضاد) الشجري- فاء- مع (اللام) الجانبي إلا على باب (ضرب).
9. لا يقع (ص) الاحتكاكي المهموس المطبق فاء مع (ن) المتوسط المجهور الخيشومي.
10. لا يتصرف (الزاي) المجهور- فاء- مع (اللام) الجانبي إلا على باب (ضرب).
11. لا يقع (ت) الانفجاري المهموس فاءً للمضعف مع (ن) المتوسط الخيشومي المجهور.
12. لا يتصرف (ط) الانفجاري المجهور المطبق- فاء- مع (ن) الخيشومي المجهور إلا على (ضرب).
13. لا يتصرف (د) الانفجاري المجهور المنفتح- فاء- مع (ن) إلا على (نصر).
14. لا يتصرف (الذال) الانفجاري المجهور- فاء- مع (ل) الجانبي إلا على (ضرب).
15. لا يتصرف (ظ) الاحتكاكي المجهور المطبق- فاء- مع (ن) الخيشومي إلا على (نصر).
16. لا يتصرف (الذال) الاحتكاكي المجهور المنفتح- فاء- مع (النون) الخيشومي إلا على (ضرب).
17. لا يقع (الثاء) الاحتكاكي المهموس فاء مع (النون) المتوسط الخيشومي المجهور.
18. لا يتصرف (الثاء) الاحتكاكي المهموس- فاء- مع (اللام) الجانبي إلا على (نصر).
19. لا يتصرف (ظ) الاحتكاكي المجهور المطبق- فاء- مع (ل) الجانبي إلا على (فتح).
20. يتصرف (الميم) المتوسط الخيشومي- فاء- مع (النون) المتوسط على (نصر).
21. يتصرف (الباء) الانفجاري المجهور- فاء- مع (النون) المتوسط على (ضرب).
22. لا يتصرف (الفاء) الاحتكاكي المهموس- فاء- مع (ن) الخيشومي إلا على (نصر).
23. لا يتصرف (الفاء) الاحتكاكي المهموس- فاء- مع (ر) المكرر إلا على (ضرب).

ثانياً : ارتباط مخرج فاء المضعف مع حيز عينه ولامه :

I - فاء المضعف من الأصوات الذلقة المتوسطة المجهورة (ر- ل- ن):

1. لا يقع (ر) المكرر المجهور فاءً مع أصوات الحلق إلا مع (خ) المهموس المستعلي.

2. لا يقع (النون) الخيشومي المجهور فاءً للمضعف مع صوتي أقصى الحلق (أ-هـ).
3. ولا يقع (النون) الخيشومي المجهور فاءً مع مجهوري وسط الحلق وأدناه: (ع-غ).
4. لا يتصرف (النون)- فاءً- مع أصوات الحلق إلا مع مهموسي وسط الحلق وأدناه: (حاء) و(خاء).
5. لا يقع (النون) الخيشومي المجهور فاءً للمضعف مع اللثويين المجهورين (ظ-ذ).
6. لا يقع (اللام) الجانبي المجهور فاءً مع أصوات الحلق المجهورة⁽⁸³⁾ (أ-ع-غ).
7. لا يقع (الراء) المجهور المكرر فاءً مع صوتي الإطباق المجهورين: النطعي (ط) واللثوي (ظ).
8. لا يقع (الراء) المكرر فاءً مع الذلقين المجهورين غير الخيشوميين (ر-ل).
9. لا يتصرف (اللام) الجانبي- فاء- مع حيز اللهاة والحنك (ق-ك) إلا على (نصر).
10. لا يتصرف (الراء) المكرر- فاء- مع الشجريات (ج-ش-ض) إلا على (نصر).
11. لا يتصرف (النون) المتوسط الخيشومي المجهور- فاء- مع الأصوات الشجرية (ج-ش-ض) إلا على باب (ضرب).
12. لا يتصرف (اللام) الجانبي- فاء- مع صوتي اللهاة والحنك (ق-ك) إلا على (نصر).
13. لا يتصرف (اللام)- فاء- مع الأصوات الأسلية (ص-س-ز) إلا على (نصر).
14. لا يتصرف (ر) المكرر- فاء- مع الأسليين المهموسين (ص-س) إلا على (نصر).
15. لا يتصرف (اللام) الجانبي مع النطعيين المنفتحين (ت-د) إلا على باب (نصر).
16. لا يتصرف (الراء)- فاءً- مع اللثويين المنفتحين (ث-ذ) إلا على باب (ضرب).
17. لا يتصرف (اللام) الجانبي- فاء- مع اللثويين (ظ-ث) إلا على (نصر).
18. لا يتصرف (ل) الجانبي- فاء- مع الأصوات الشفهية (ف-ب-م) إلا على (نصر).

II - عين المضعف ولامه من الأصوات الذلقية المتوسطة المجهورة (ر-ل-ن):

1. لا يمتنع تصرف أي من الأصوات الحلقية- فاءً- مع الأصوات الذلقية (ر-ل-ن).
2. لا يتصرف (هاء) الحلقي المهموس- فاءً- إلا على (ضرب) مع الجانبي (ل)، والخيشومي (ن).
3. يتصرف (العين) وسط الحلقي المجهور- فاءً- على البابين (نصر) و(ضرب) مع الذلقيات الثلاثة (ر-ل-ن).
4. لا يتصرف (الزاي) الأسلي الاحتكاكي المجهور- فاء- مع المكرر (ر) والخيشومي (ن) إلا على (نصر).
5. لا يتصرف (الطاء) اللثوي الاحتكاكي المجهور المطبق- فاءً- مع (ر) المكرر و(ن) الخيشومي إلا على باب (نصر).
6. لا يتصرف (الذال) اللثوي الاحتكاكي المجهور المنفتح- فاء- مع (ل) الجانبي و(ن) الخيشومي إلا على باب (ضرب).
7. يتصرف (الغين) أدنى الحلقي المجهور المستعلي- فاءً- مع الذلقين (ر-ل) على (نصر) و(فتح).

(83) ذكر سيبويه أن الهمزة من الأصوات المجهورة : الكتاب- ج4- ص434.

8. لا يتصرف (ك) الحنكي المهموس- فاء- مع (ر) المكرر، (ن) الخيشومي إلا على (نصر).
9. يتصرف (الباء) الشفهي المجهور- فاء- مع الذقنين غير الخيشوميين (ر- ل) على الأبواب (نصر) و(ضرب) و(فتح).
10. يتصرف (الميم) الشفهي الخيشومي- فاء- مع الذقنين غير الخيشوميين (ر- ل) على البابيين (نصر) و(فتح).

ثالثا : ارتباط حيز فاء المضعف مع مخرج عينه ولامه :

I - فاء المضعف من الأصوات الذلّقية المتوسطة المجهورة (ر- ل- ن):

1. لا تقع الأصوات الذلّقية المتوسطة المجهورة (ر- ل- ن) فاءً مع (العين) وسط الحلقي المتوسط المجهور.
2. لا تقع الذلقيات المجهورة (ر- ل- ن) فاءً مع (الغين) أدنى الحلقي المجهور المستعلي.
3. لا يقع الذلقيان المجهوران (ر- ن) فاءً مع (الهاء) أقصى الحلقي المهموس.
4. لا يتصرف الذلقيان غير الخيشوميين (ر- ل)- فاء- مع (الخاء) أدنى الحلقي المهموس المستعلي إلا على باب (نصر).
5. لا يتصرف الذلقيان (ر- ن)- فاء- مع (القاف) اللهوي المجهور إلا على (ضرب).
6. لا يتصرف (اللام) و(النون)- فاء- مع (الجيم) الشجري المجهور إلا على (ضرب).
7. لا تتصرف الذلقيات (ر- ل- ن)- فاء- مع (ت) النطعي المهموس إلا على (نصر).
8. لا يتصرف الذلقيان غير الخيشوميين (ر- ل)- فاء- مع (الذال) النطعي المجهور المنفتح إلا على (نصر).
9. لا يتصرف (ل- ن)- فاء- مع (ط) النطعي المجهور المطبق إلا على (ضرب).
10. لا يتصرف الذلقيان غير الخيشوميين (ر- ل)- فاء- مع (ب) الشفهي الانفجاري إلا على (نصر).

II - عين المضعف ولامه من الأصوات الذلّقية المتوسطة المجهورة (ر- ل- ن):

1. لا تتصرف الحلقيات (أ- ه- ح) فاء- مع (ن) الخيشومي إلا على (ضرب).
2. لا يتصرف صوتا اللهاة والحنك (ق-ك)- فاء- مع (ل) الجانبي إلا على (ضرب).
3. لا يتصرف (ق-ك)- فاء- مع (النون) الخيشومي إلا على باب (نصر).
4. لا يتصرف (ج-ض) الشجريان المجهوران- فاء- مع (ر) المكرر إلا على (نصر).
5. لا يتصرف الأسليان الاحتكاكيان (س-ز)- فاء- مع (ل) الجانبي إلا على (ضرب).
6. لا يتصرف الأسليان الاحتكاكيان (س-ز)- فاء- مع (ن) الخيشومي إلا على (نصر).
7. لا يتصرف الشفهيان (ف-م)- فاء- مع (النون) الخيشومي إلا على باب (نصر).

رابعا : ارتباط حيز فاء المضعف مع حيز عينه ولامه :

I - فاء المضعف من الأصوات الذلّقية المتوسطة المجهورة (ر- ل- ن):

1. لا يقع أحد الذلقيات (المجهورة) (ر- ل- ن) فاءً مع الحلقيات المجهورة (أ- ع- غ).
2. لا يقع الذلقيان المجهوران: (اللام) الجانبي و(النون) الخيشومي فاءً مع أصوات الحلق إلا مع (الحاء) و(الخاء) المهموسين من وسط الحلق وأدناه.

3. لا يتصرف الذلقيان غير الخيشوميين (ر-ل)- فاء- مع الأسليين المهموسين (ص-س) إلا على (نصر).
4. لا يتصرف الذلقيان غير الخيشوميين (ر-ل)- فاء- مع النطعيين المنفتحين (ت-د) إلا على (نصر).
5. لا يقع (اللام) الجانبي و(النون) الخيشومي فاءً مع الذلقيات الثلاثة: (ر-ل-ن).
6. لا يقع أحد الأصوات الذلقية (ر-ل-ن) فاءً مع (الراء واللام) غير الخيشوميين.

II - عين المضعف ولامه من الأصوات الذلقية (ر-ل-ن):

1. لا يمتنع تصرف أي من الأصوات الحلقية- فاء- مع الأصوات الذلقية (ر-ل-ن).
2. يتصرف الأسليان المنفتحان (س-ز)- فاء- مع (الراء والنون) على (نصر).
3. يتصرف النطعيان (ط-ت)- فاء- مع (الراء واللام) على البابين (نصر) و(ضرب).
4. لا يقع (اللام والنون) الذلقيان فاءً مع الأصوات الذلقية الثلاثة (ر-ل-ن).
5. يتصرف الشفهيان المجهوران (ب-م)- فاء- مع الذلقيين المجهورين غير الخيشوميين (ر-ل) على بابي (نصر) و(فتح).

بهذا يكون البحث قد حقق أهدافه بالإجابة عن التساؤلات اللغوية الواردة في مطلعته

:

- فقد تبين أثر حيز صوتي الفعل المضعف في ورود الفعل على باب صرفي بعينه.
- كما اتضح أن أحياز الأصوات ومخارجها كانت عاملاً حاكماً في اختيار الباب الصرفي على لسان العرب القدامى.
- وعلى مستوى الأصوات الذلقية رأينا أن الاتحاد في مخرج أصوات الحيز هو العامل الحاكم في إثارة الفعل باباً صرفياً بعينه.

- كما أمكن، من خلال البحث اللغوي، تلمس بعض القواعد التي تحكم السلوك الصرفي للفعل المضعف، الذي يكون أحد صوتيه من حيز الأصوات الذلقية على النحو التالي:

1. لا يتصرف (ر) المكرر- فاء- مع حيز شجر الفم (ج-ش-ض) إلا على (نصر).
2. لا يتصرف (ن) الخيشومي- فاء- مع الشجريات (ج-ش-ض) إلا على (ضرب).
3. تتناظر الذلقيات المجهورة (ر-ل-ن)- فاء- مع الحلقيات المجهورة (أ-ع-غ).
4. لا تقع الأصوات الذلقية (ر-ل-ن) فاءً مع (الهمزة) و(الهاء) أقصى الحلقيين.
5. لا تقع الأصوات الذلقية المجهورة (ر-ل-ن) فاءً مع (العين) وسط الحلقى المجهور، ومع (الغين) أدنى الحلقى المجهور المستعلي.
6. لا يتصرف (النون) المجهور- فاء- على باب (نصر) منفرداً إلا مع (التاء) النطعي المهموس المنفتح، و(الفاء) الشفهي المهموس المنفتح.
7. لا يتصرف (النون) المجهور- فاء- مع الأصوات المجهورة من الأسلّة ونطع الغار: (ز-ط-د) إلا على باب (ضرب).
8. يتصرف (النون) المتوسط الخيشومي المجهور- فاء- على باب (ضرب) منفرداً مع المستعليين من الحلق واللهاة (خ-ق)، والأصوات الشجرية (ج-ش-ض)، والشفهي المجهور (ب).

9. لا يتصرف (ل) - فاء- مع أحياء: اللهاة والحنك، والأسلة، والشفنتين إلا على (نصر).
10. لا يتصرف (ل) الذلقي الجانبي- فاء- على (فتح) إلا مع (ذ) اللثوي المجهور المنفتح.
11. لا يتصرف (ل) الذلقي الجانبي المجهور- فاء- على (ضرب) إلا مع الحلقي المهموس (ح) والمجهورين: (ج) الشجري، و(ط) النطعي.
12. يتصرف (اللام) الذلقي الجانبي- فاء- على (نصر) منفردا مع الحلقيين المهموسين الاحتكاكيين (هـ- خ)، ومع (ت- د) النطعيين الانفجاريين المنفتحين، و(ظ- ذ) اللثويين الاحتكاكيين المجهورين.
13. (الراء) الذلقي المكرر هو الصامت الوحيد الذي لا يتنافر مع أصوات حيزه: فيتصرف- فاء- مع (النون) الذلقي الخيشومي على باب (ضرب)، في الفعل (رنّ).
14. حين يقع (الراء) الذلقي المكرر- فاء- يتصرف على باب (نصر) دائما⁽⁸⁴⁾.
15. لا يتصرف (الراء) الذلقي المكرر- فاء- إلا على (نصر) مع حيزي الأصوات الشجرية (ج- ش- ض)، والأسلية (ص- س- ز)⁽⁸⁵⁾.
16. يتصرف (الراء) الذلقي المكرر- فاء- على (نصر) منفردا مع النطعيين الانفجاريين المنفتحين (ت- د)، والشفهي الانفجاري المجهور(ب).
17. لا يتصرف (الراء) الذلقي المكرر- فاء- إلا على (ضرب) مع (القاف) اللهوي الانفجاري المجهور المستعلي، و(ث) و(ذ) اللثويين الاحتكاكيين المنفتحين المستقلين، و(النون) الذلقي الخيشومي المجهور.
18. يتمثل تصرف صوتي اللهاة والحنك (ق- ك)- فاء- مع (ل) الجانبي و(ن) الخيشومي فلا يتصرفان مع (ل) إلا على (ضرب)، ولا يتصرفان مع (ن) إلا على (نصر).
19. يتمثل تصرف الأسليين المنفتحين (س- ز)- فاء- مع (ل) الجانبي و(ن) الخيشومي: فلا يتصرفان مع (ل) إلا على (ضرب)، ولا يتصرفان مع (ن) إلا على (نصر).
20. لا يقع (النون) الخيشومي فاء مع (أ- هـ) من أقصى الحلق، و(ع- غ) الحلقيين الاحتكاكيين المجهورين، و(ك) الحنكي الانفجاري المهموس، و(ظ- ذ) اللثويين الاحتكاكيين المجهورين، فضلا عن الأصوات الذلقية (ر- ل- ن).

ويبقى أن يختبر البرنامج الحاسوبي هذه القواعد؛ ليؤكد الصحيح منها، ويضبط غير الدقيق، ويضيف ما لم يستطع العقل البشري حصره أو الإحاطة به.

(84) تتصرف الأفعال (رك- رف- رم) أيضا على باب (ضرب)، مع اختلاف الدلالة على البابين.

(85) يتصرف الفعل (رّ) أيضا على باب (ضرب)، مع اختلاف الدلالة على البابين.

مصادر البحث ومراجعته

- الاسترابادى (رضى الدين محمد بن الحسن) :
شرح شافية ابن الحاجب (تحقيق الزفزاف)- دار الكتب العلمية – بيروت 1982.
ابن جنى (أبو الفتح عثمان) :
- الخصائص – تحقيق النجار – ط 2 – دار الهدى – بيروت (ب ت).
- سر صناعة الإعراب- تحقيق هنداوي – ط 2 – دار القلم – دمشق 1993.
الخليل بن أحمد الفراهيدي:
- كتاب العين – تحقيق عبد الله درويش – بغداد 1967.
- كتاب العين- ت مهدي المخزومي، إبراهيم السامرائي- دار الرشيد- بغداد 1982.
ابن درستويه (عبد الله بن جعفر) :
تصحيح الفصيح وشرحه – مجلس الشئون الإسلامية- القاهرة 1419هـ.
ابن دريد (أبو بكر محمد بن الحسن):
جمهرة اللغة – دار صادر – بيروت.
السرقسطى (أبو عثمان سعيد بن محمد المعافى) :
كتاب الأفعال – الهيئة العامة لشئون المطابع الأميرية- القاهرة 1992.
سيبويه (أبو بشر عمرو بن عثمان بن قنبر) :
الكتاب ج 4 – ط 2- مكتبة الخانجي- القاهرة 1982.
السيوطى (عبد الرحمن جلال الدين) :
- المزهرة فى علوم اللغة وأنواعها- المكتبة العصرية- بيروت 1986.
- همع الهوامع فى شرح جمع الجوامع- تحقيق أحمد شمس الدين- ط 1- دار الكتب العلمية- بيروت 1998.
الفارابي (إسحق بن إبراهيم) :
ديوان الأدب – ت: أحمد مختار عمر- مطبوعات مجمع اللغة العربية، القاهرة 1975.
الفيروزابادى (مجد الدين محمد بن يعقوب) :
القاموس المحيط – دار الكتاب العربي – بدون تاريخ أو مكان الطبع.
ابن القطاع (الصقلى) :
أبنية الأسماء والأفعال والمصادر- ت أحمد عبد الدايم – ط دار الكتب المصرية – القاهرة 1999.

- ابن القوطية (أبو بكر محمد بن عمر) :
كتاب الأفعال- الطبعة الثانية- مكتبة الخانجي- القاهرة 1993.
المبرد (أبو العباس محمد بن يزيد) :
المقتضب – المجلس الأعلى للشئون الإسلامية – القاهرة 1399.
ابن منظور (جمال الدين أبو الفضل محمد) :
لسان العرب – دار المعارف – القاهرة 1981.

وفاء كامل فايد:

- أثر تجاوز صوتي الفعل الثلاثي المضعف في بابہ الصرفي: دراسة في حيزي الحلق والشفنتين: مؤتمر مجمع اللغة العربية بالقاهرة، عام 2009.
- أثر تجاوز صوتي الفعل الثلاثي المضعف في بابہ الصرفي: دراسة في الأحياز الوسطية: مؤتمر مجمع اللغة العربية بالقاهرة، أبريل 2010.
- الأفعال المضعفة وأبوابها الصرفية " : المجلة العربية للعلوم الإنسانية- جامعة الكويت- ع 74- س19، عام 2001.
- الباب الصرفي للفعل المضعف وأحياز أصواته: دراسة في الأحياز الوسطية والذلقية، بحوث الكتاب التذكري (ثمرات الامتتان)- مكتبة الخانجي، ط1- القاهرة 2002.
- تراكب الأصوات في الفعل الثلاثي الصحيح -عالم الكتب- القاهرة 1991.
- مدى ارتباط الفعل الثلاثي الصحيح بالمضارع المفتوح العين- (دراسة إحصائية على القاموس المحيط)، العدد 58: مجلة كلية الآداب- ج القاهرة: مارس 1993.
ابن يعيش (موفق الدين يعيش بن علي) :
شرح المفصل- عالم الكتب- بيروت، ب ت.

Representing Arabic Documents Using Controlled Vocabulary Extracted from Wikipedia

Mohamed I. Eldesouki^{*1}, Waleed M. Arafa^{*2}, Kareem Darwish^{**3}, Mervat H. Gheith^{*4}

** Department of Computer and Information Sciences, Institute of Statistical Studies and Research, Cairo University
5 Dr. Ahmed Zewel Street, Orman, Giza, Egypt*

¹disooqi@ieee.org

²waleed_arafa@hotmail.com

⁴mervat_gheith@yahoo.com

*** Qatar Computing Research Institute, Qatar Foundation
Al-Nasr Tower A, 21st Floor, Doha, Qatar*

³kareem@darwish.org

Abstract— One of the key aspects in Information Retrieval is the way to represent documents to be retrieved. Some systems, which use documents' keywords only to represent the documents, might neglect indexing of words that of less meaning. Other systems try to choose the most representative keywords for their documents. The same set of keywords could be used with different levels of analysis to provide different representations for the documents.

In this work, we used Arabic Wikipedia project as source of controlled vocabulary and use this controlled vocabulary for indexing documents. Our technique is very close to the work of Eldesouki [15]. However, instead of using ids to represent the documents, we use the terms themselves to represent each document.

We examined normalizing the documents before applying our technique. Furthermore, we examined stemming the documents before, after, and while applying our technique. The mean average precision of our technique outperforms light10 stemmer. Although the difference is not statistically significant, our technique shows that many terms produced from just stemming are not significant in representing the documents.

Furthermore, using our technique dramatically decreases the size of the index. Experiments show that our technique reduces about 47.5% of the size of the index build from applying light10 stemmer.

1 INTRODUCTION

One of the key aspects in Information Retrieval is the way to represent documents to be retrieved, a so-called logical view of the document. Some systems use the full set of words to represent documents, whereas others use subset of the words to represent documents in the system. The representing of a document could be viewed as a continuum in which it might shift from a full text representation to a higher level representation specified by a human subject [6].

Some systems, which use documents' keywords only to represent the documents, might neglect indexing of words that of less meaning. Pronouns, prepositions, and conjunctions are the typical examples of such words. Some systems keep a list of such words (stopwords list) to prevent from indexing them. Other systems try to choose the most representative keywords for their documents based on factors such as the frequencies of such keywords, their spread over a single document and others.

The same set of keywords could be used with different levels of analysis to provide different representations for the documents. Using different levels of analysis helps to overcome the problem of matching between two sequences of characters.

Different techniques have been developed to overcome the difficulties for matching process including normalization process, stemming process, morphological analysis process, n-gram for words, using ontologies, etc.

In this work, we investigate representing documents using terms of controlled vocabulary extracted from Arabic Wikipedia project. Using this controlled vocabulary, we use a special n-gram algorithm to identify the entities within the text. We further examined using stemming technique before and after applying our technique. The results are compared to other stemming techniques [14].

The rest of the paper is organized as follows: section 2 presents the previous work; section 3 presents the methods of using Wikipedia as source of concepts; section 4 briefly introduces the different disambiguation techniques that have been examined,

section 5 describes the experiment carried out to evaluate the stemmers. Results and discussions are provided in section 6 and conclusion is derived in section 7.

2 PREVIOUS WORK

Abu El-Khair [1] has examined three examples of Arabic stopwords lists for their effectiveness in information retrieval system.

After morphologically analyzing text, Mansour and his companions [25] assign weights for each terms of a document and then sort the terms in descending order by weight to help selecting them later. The weight of a word depends on three factors; the frequency of occurrence in a document, the count of stem words for that word, and on the spread of the word in the document.

In his work, Mohamed Eldesouki [15] has used the Arabic Wikipedia project to represent each document as a set of ids. Each id represents a single entity in the text of the document. Many forms and variants are encoded within these ids (such as synonyms, acronyms, words with different affixes and different morphological variations). Furthermore, the representation using these ids avoids the problem of polysemy since words with different senses assigned different ids. However, two issues constitute the main obstacles for his approach; the first one is the use of word sense disambiguation technique to disambiguate the right sense of terms that has multiple senses. The other problem is the immature nature of the Arabic Wikipedia project which is yet not contain the sufficient amount of variants and forms to represent all the (or even the majority) of the terms variants

Al-Kharashi [4] tried to use dictionaries of roots and stems, built manually, for each word to be indexed. The roots and stems extracted from a very small collection of text.

Arabic morphological Analyzers have been used to obtain the roots of the words automatically to be indexed. A lot of analyzers exist in that time have been used and evaluated; for example Khoja Morphological Analyzer [19], Tim Buckwalter morphological analyzer 1.0 [24], ALPNET morphological analyzer [7], and Sebawai [10].

A controversial issue at that time was whether to use roots or stems as terms for indexing. Several studies have claimed that roots outperform stems [4], [17], [2] and [9]. However, most of the resent studies found that using stems as index terms outperform roots; [5], [21], [11], [22], [28], [12]. The reason that the former researchers, that found roots better than stems for IR tasks, have done their experiment on small collections of text which is not enough for evaluation.

Using the TREC-2001 Arabic corpus [23], experiments reveal that roots are not suitable because Arabic consists of a few thousands of roots. Analyzing each word to its root would conflate many words of different meaning to the same class. For example, the Arabic words for office, book, Library, writer, and letter have same root.

After TREC Arabic cross-language Information retrieval tracks (CLIR) [16], researchers have directed their research to use stems as index terms. They developed a lot of stemmers to handle Arabic Language in IR context. Many studies have been conducted in stemming techniques; [11], [5], [21], [8], [22], [3], [26], [20], [27], and [13].

3 OUR TECHNIQUE

Our technique is very close to the work of Eldesouki [15]. However, instead of using ids to represent the documents, we use the terms themselves to represent each document. In other words, we use the terms to represent the document if and only if they exist in Arabic Wikipedia as articles' titles. The main idea behind this technique is assuming that noun phrases are more representative than verbs, adjectives and adverbs. And we use Wikipedia as a source of the noun phrases to use as a controlled vocabulary.

We overcome the problem of variants limitation in Arabic Wikipedia by using the best stemming technique which is the light10 stemmer to stem the text; to the best of our knowledge [14].

4 TERMS IDENTIFICATION

The term detection or identification task goes as follows: the document is firstly tokenized. The document is then processed to generate word n-grams. The n-gram generation process differs from the usual way of producing n-gram; See Algorithm in Table I. While the system generates n-grams, it tries to match the n-gram to the variants of each different article's titles that have extracted from Wikipedia. The size of the n-gram, n, is equal to longest variant length. Although, there is small likelihood to produce wrong phrases, the customized method for generating n-gram has the advantage of reducing ambiguity by trying to detect longer phrases first.

Our technique could be used with other text processing technique such as normalization, stemming or even morphological analysis. Our technique could be applied before or after these text processing techniques.

TABLE I
ALGORITHM OF TERMS IDENTIFICATION

Input: <i>TokensQ</i> (queue of all document tokens), <i>synDic</i> (variants dictionary), <i>n</i> (size of n-gram)
Output: list of tokens of identified terms in the document
Algorithm:
1) If <i>TokensQ</i> size = 0, then return;
2) Else If <i>TokensQ</i> size $\geq n$, Choose first <i>n</i> tokens from the <i>TokensQ</i> into <i>nList</i> (a list of n-gram size).
3) Else, choose <i>all</i> tokens from the <i>TokensQ</i> into <i>nList</i> .
4) Constitute a n-gram by concatenating all the tokens in <i>nList</i> .
5) Try to find the term in the variants dictionary
6) If (variant found in <i>synDic</i>)
a) Consider the tokens of the variant to be indexed
b) Empty <i>nList</i> and dequeue the tokens of the term from the <i>TokensQ</i>
c) Go to step 1.
7) Else (the term has no corresponding in <i>synDic</i>)
a) Then remove one token from the end of <i>nList</i> .
b) Check the size of <i>nList</i> after removal
i) If number of tokens that exist in <i>nList</i> = 0, dequeue the last removed token from <i>TokenQ</i> and go to step 1.
ii) If number of tokens that exist in <i>nList</i> > 0, then go to step 4.

5 WIKIPEDIA AS SOURCE OF CONTROLLED VOCABULARY

Wikipedia is a free encyclopedia that maintains topics and subjects that covers many areas of knowledge. Articles of Wikipedia usually describe ideas or define specific terminologies. Wikipedia is not a dictionary; it doesn't contain general words.

We use a controlled vocabulary built from the titles of Wikipedia's article to represent documents. The key idea behind our technique is that instead of using a general dictionary or lexicon to represent document, we use a set of constantly-increasing terminologies to represent the documents.

The continuous growth of the Wikipedia project makes it a good source of a controlled vocabulary. Due to collaboration work of volunteers, the Wikipedia grows constantly and rapidly. This gives it more advantage than other resources which is fixed in size such as Arabic WordNet. The Wikipedia produces a database dump every 15 days. This makes the Wikipedia reflects the reality and makes it up-to-date.

We used Arabic Wikipedia project as source of the controlled vocabulary. The controlled vocabulary has been extracted using two ways. Redirect pages and the anchors' text of interlinks between articles of Arabic Wikipedia.

6 EXPERIMENTS SETUP

The experiments measure the effect of using index terms produced by our technique to improve retrieval effectiveness of the information retrieval system.

As we mentioned earlier, our technique could be used in existence of other text processing steps such as normalization and stemming. We examined normalizing the documents before applying our technique. Furthermore, we examined stemming the documents before, after, and while applying our technique. We choose the light10 stemmer to stem the text since it is the outperforming stemmer [14]. We experiment using a controlled vocabulary extracted from only redirect pages and from both redirect pages and the anchors' text of interlinks between articles. Each experiment is conducted with and without relevance feedback.

The results of our techniques are compared with stemming techniques, since they outperform the other techniques for processing Arabic text [14].

We have used TREC-2001 Arabic corpus for evaluation. TREC-2001 Arabic corpus, also called the AFP_ARB corpus, consists of 383,872 newspaper articles in Arabic from Agence France Presse. This fills up almost a gigabyte in UTF-8 encoding as distributed by the Linguistic Data Consortium. There were 25 and 50 topics used in 2001 and 2002 respectively with relevance judgments, available in Arabic, French, and English, with Title, Description, and Narrative fields. We used the Arabic titles and descriptions as queries of the 75 topics in the experiments.

For all the experiments, we used the Lemur language modeling toolkit [30], which was configured to use Okapi BM-25 term weighting with default parameters and with and without blind relevance feedback (the top 50 terms from the top 10 retrieved documents were used for blind relevance feedback). To observe the effect of alternate indexing terms, mean average precision, MAP, was used as the measure of retrieval effectiveness. To determine if the difference between results was statistically significant, a paired t-test [18] and Wilcoxon sign test [29] have been used with p values less than 0.05 as indication for significance.

As a requirement for Arabic text to be indexed with Lemur toolkit, corpus and topics have been converted to CP1256 encoding. Then a normalization step was performed. The encoding conversion and normalization steps were conducted on both text collection and the topics where queries were extracted. We applied our technique to the topics as required.

In order to be able to compare the retrieval performance with the light stemmers mentioned in [14], the same experiment parameters have been used for current work.

7 RESULTS AND DISCUSSION

Table II shows the results of applying our technique after normalizing the documents as well as the results of stemming the documents before, after and while applying our technique.

TABLE II
EXPERIMENTS USING OUR TECHNIQUE (BOTH REFERS TO REDIRECT PAGES AND ANCHORS' TEXT)

	Use Normalized Text		Stem Text					
			Before		Through		After	
	With	Without	With	Without	With	Without	With	Without
Redirect only	0.2690	0.2296	0.3791	0.3471	0.3327	0.29	0.3327	0.2848
both	0.3056	0.2470	0.3936	0.3510	0.3521	0.2969	0.3919	0.3496

The experiments are conducted using controlled vocabulary extracted from only redirect pages and from both redirect pages and context of other articles. All experiments are conducted with and without blind relevance feedback.

The results show that using stemming before or after applying technique dramatically increases the performance of the information retrieval system. The table shows that the difference between normalizing text and stemming text before applying our technique is statistically significant where the t-test and sign test values are 0.0002 and 0.00, respectively, with query expansion and when extracting Wikipedia data using both methods.

In the other hand, using both redirect pages and anchors' text dramatically increase the performance of Information Retrieval system over using just the redirect pages.

For using stemming technique, the difference between using stemming technique before applying our technique and after applying our technique is not statistically significant with and without query expansion where t-test is 0.3837 and sign test is 0.3638 when expanding, and t-test is 0.3801 and sign test is 0.1778 when not expanding. We have to note that this result is for using "both" ways of extracting Wikipedia methods. In case of using only redirect pages, the difference between stemming after and stemming before is significant, where the t-test and sign tests are 0.005 and 0.0001, respectively when expanding, and 0.0003 and 0.000, respectively when not expanding.

Table III shows the index sizes for the different experiments. It shows that using both ways for extracting controlled vocabulary always increases the size of the index. Furthermore, stemming the documents after applying our technique gives the smallest index size.

TABLE III
THE SIZES OF INDICES FOR ALL EXPERIMENTS

	Use Normalized Text		Stem Text					
			Before		Through		After	
	With	Without	With	Without	With	Without	With	Without
Redirect only	335 MB		471 MB		480 MB		308 MB	
both	424 MB		528 MB		541 MB		382 MB	

Table IV is intended for comparing between our technique, light10 stemmer, and the technique in [15] in terms of performance and index sizes. The table shows that although our technique slightly improves the performance over light10 stemmer, the

different is not statistically significant. However, this could be used as an indication that, when using only stemming, many terms indexed are not important in representing the documents.

Although, our technique adds a burden to the information retrieval system (since it adds another task before or after stemming the text), using our technique dramatically decreases the size of the index by about 47.5%.

TABLE IV
COMPARISON BETWEEN INDEX BUILD FROM THE COLLECTION AFTER APPLYING JUST LIGHT10 STEMMING, OUR TECHNIQUE, AND THE TECHNIQUE IN [15]

	With Query Expansion	Without Query Expansion	Index Size
Technique of [15]	0.3394	0.3813	631 MB
Light10	0.3914	0.3489	727 MB
Our technique (stem first)	0.3936	0.3510	528 MB
Our technique (stem later)	0.3919	0.3496	382 MB

8 CONCLUSIONS

The mean average precision of our technique outperforms light10 stemmer. Although the difference is not statistically significant, our technique shows that many terms produced from just stemming are not significant in representing the documents.

Furthermore, using our technique dramatically decreases the size of the index. Experiments show that our technique reduces about 47.5% of the size of the index build from applying light10 stemmer.

REFERENCES

- [1] Abu El-Khair I., 2006, Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study, *International Journal of Computing & Information Sciences*, pages 119-133.
- [2] Abu-Salem, H., Al-Omari, M., and Evens, M. Stemming methodologies over individual query words for Arabic information retrieval. *Journal of the American Society for Information Science (JASIS)*, 50 (6), pp. 524-529, 1999.
- [3] Al-Ameed k. Hayder, Al-Ketbi O. Shaikha, Al-Kaabi A. Amna, Al-Shebli S. Khadija, Al-Shamsi F. Naila, Al-Nuaimi H. Noura, Al-Muhairi S. Shaikha, Arabic Light Stemmer: A new Enhanced Approach, *The second international conference on innovations technology (IIT'05)*, 2005.
- [4] Al-Kharashi, I. and Evens, M. W. Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *Journal of the American Society for Information Science (JASIS)*, 45 (8), pp. 548-560, 1994.
- [5] Aljlal, M., & Frieder, O. On Arabic search: Improving the retrieval effectiveness via light stemming approach. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, Illinois Institute of Technology (pp. 340-347). New York: ACM Press.2002.
- [6] Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. AddisonWesley.
- [7] Beesley, K. R. Arabic finite-state morphological analysis and generation. In *COLING-96: Proceedings of the 16th international conference on computational linguistics*, vol. 1, pp. 89-94, 1996.
- [8] Chen, A., and Gey, F. Building an Arabic stemmer for information retrieval. In *TREC 2002. Gaithersburg: NIST*, pp 631-639, 2002.
- [9] Darwish, K., Doermann, D., Jones, R., Oard, D., and Rautiainen, M. *TREC-10 experiments at Maryland: CLIR and video*. In *TREC 2001*. Gaithersburg: NIST, 2001.
- [10] Darwish, K. Building a shallow morphological analyzer in one day. *ACL 2002 Workshop on Computational Approaches to Semitic languages*, July 11, 2002.
- [11] Darwish, K. and Oard, D.W. CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval. In *TREC 2002. Gaithersburg: NIST*, pp 703-710, 2002.
- [12] Darwish K., Hassan H., and Emam O., Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 25-30, Ann Arbor, June 2005.
- [13] El-Beltagy S., Rafea A.. A FRAMEWORK FOR THE RAPID DEVELOPMENT OF LIST BASED DOMAIN SPECIFIC ARABIC STEMMERS, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 2009.
- [14] Eldesouki M., Arafa W. and Darwish K. Stemming techniques of Arabic Language: Comparative Study from the Information Retrieval Perspective. *The Egyptian Computer Journal* , Vol. 36 No. 1, June 2009.
- [15] Eldesouki M., Arafa W., Darwish K., and Gheith M., Using Wikipedia for Retrieving Arabic Documents, *Proceedings of Arabic Language Technology International Conference*, October 2011.
- [16] Gey, F. C. and Oard, D. W. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French, or Arabic queries. In *TREC 2001. Gaithersburg: NIST*, 2002.
- [17] Hmeidi, I., Kanaan, G. and M. Evens (1997) Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. *Journal of the American Society for Information Science*, 48/10, pp. 867-881.
- [18] Hull, D. Using Statistical Testing in the Evaluation of Retrieval Performance. In *Proceedings of the 16th ACM/SIGIR Conference*, pages 329-338, 1993.
- [19] Khoja, S. and Garside, R. Stemming Arabic text. *Computing Department, Lancaster University*, Lancaster, 1999.
- [20] Kadri, Y., and Nie, J. Y. (2006), Effective stemming for Arabic information retrieval". The challenge of Arabic for NLP/MT Conference, The British Computer Society. London, UK.
- [21] Larkey, Leah S., Ballesteros, Lisa, and Connell, Margaret. (2002) Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In *Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002)*, Tampere, Finland, August 11-15, 2002, pp. 275-282.
- [22] Larkey, S. L., Ballesteros, L., and Connell, E. M. (2005), Light stemming for Arabic information retrieval. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*.
- [23] LDC, Linguistic Data Consortium. LDC2001T55, 2001. Available from: <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T55> (accessed 1 August 2011)
- [24] LDC, Linguistic Data Consortium. Buckwalter Morphological Analyzer Version 1.0, LDC2002L49, 2002. Available from: <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49>, (accessed 1 August 2011)

- [25] Mansour N., Ramzi A. Haraty, Walid Daher, and Manal Hourri. 2008. An auto-indexing method for Arabic text. *Inf. Process. Manage.* 44, 4 (July 2008), 1538-1545.
- [26] Nwesri A., S.M.M Tahaghoghi, Falk Scholer, Stemming Arabic Conjunctions and Prepositions, In Mariano Consens and Gonzalo Navarro (eds.), *Lecture Notes in Computer Science - Proceedings of the Twelfth International Symposium on String Processing and Information Retrieval (SPIRE'2005)*, Buenos Aires, Argentina, 3772:206-217, November 2-4,2005.
- [27] Nwesri A., S.M.M. Tahaghoghi and Falk Scholer, Arabic Text Processing for Indexing and Retrieval, *Proceedings of the International Colloquium on Arabic Language Processing*, Rabat, Moroc, 18-19 June, 2007.
- [28] Taghva, K., Elkoury, R., and Coombs, J. Arabic Stemming without a root dictionary. 2005.
- [29] Wonnacott, R., Wonnacott, T. *Introductory Statistics*, John Wiley & Sons, Fourth Edition,1990.
- [30] Lemur Project Website: <http://www.lemurproject.org/> (accessed 1 August 2011)

Text Generation Model from Rich Semantic Representations

Dalia Sayed¹, Mostafa Aref², Ibrahim Fathy³

*Department of Computer science,
Faculty of Computer and Information Sciences,
Ain-Shams University, Cairo, Egypt.*

¹dalia_sayed_43@hotmail.com

²aref_99@yahoo.com

³ibrahim.moawad@heic.eg

Abstract - Natural Language generation (NLG) is one of the oldest subfield of language processing when computers were able to understand only the most unnatural of command languages they were spitting out natural texts. NLG focuses on the generation of written texts in natural languages from some underlying semantic representation of information. This paper proposes a new model to generate Multiple English texts from semantic graph. This semantic graph uses semantic graph representation called "Rich Semantic Graph" (RSG). RSG is a new ontology-based representation to generate a unified semantic representation for different NLP applications like machine translation, text summarization, and information retrieval. The model uses the WordNet ontology to generate multiple texts according to the word synonyms. Also, the model enables users to determine the output text style by selecting one of two writing styles (Cause/Effect and Description/Narration). NLG consists of five tasks: text planning, sentence planning, surface realization, Writing style selection and evaluation. We are going to evaluate the generated text with respect to text coherence and readability measurements.

1 INTRODUCTION

Natural language generation (NLG) is a subfield of natural language processing. Language understanding is somewhat like counting from one to infinity. Language generation is like counting from infinity to one. NLG focuses on the generation of written texts in natural languages from some underlying semantic representation of information. This representation generally comes from databases or knowledge sources. Accomplishing this goal may be envisioned for a number of different purposes. Including standardized and/or multi-lingual reports, summaries, machine translation, dialogue applications, and embedding in multi-media and hypertext environments. Consequently, the automated production of language is associated with a large number of highly diverse tasks whose appropriate orchestration in high quality poses a variety of theoretical and practical problems. Relevant issues include content selection, text organization, and production of referring expressions, aggregation, lexicalization, and surface realization, as well as coordination with other media. In this paper; section 2 presents brief background and review of the related work. Section 3 illustrates the NLG conceptual view. Section 4 discusses the NLG model phases. Section 5 illustrates how NLG model work through a real example. Finally section 5 concludes the paper and presents future work.

2 BACKGROUND AND RELATED WORK

The mission of generating text in natural language from data which is not linguistic by its nature can be divided into a series of sub tasks. Most NLG systems share a high similarity in the tasks performed, and in the division of the overall work into sub tasks. These tasks are performed by modules arranged as a pipeline, so the output of each module is the input of the next one. These modules are not totally detached in all implementations of NLG systems; also the streaming of information between modules is not always linear. Nevertheless efforts have been done in the NLG community in order to define the common tasks and components needed in order to build an NLG system [**Error! Reference source not found.**]. The high level generation tasks are: Text planning ("what shall I say?") sentence planning ("why should I say it this way?") surface realization ("how shall I say it?") [1].

One of the most recent researches related to this work is Natural OWL (Ontology Web Language) [2]. It is natural language generation engine that produces descriptions of entities (e.g., items for sale, museum exhibits) and classes (e.g., types of exhibits) in English and Greek from OWL ontology. The ontology must have been annotated with linguistic and user modeling annotations that may be edited using a plug-in Protégé ontology editor. Another related research is Generating Natural Language Descriptions of Ontology Concepts [3]. Their model generates NL descriptions of classes defined in OWL ontology [3]. Attributes of classes are described in OWL by defining restrictions that apply to called properties. Properties are binary relations among ontology objects. Since OWL ontology does not contain the information necessary for lexicalization, lexical information was added by a rule-based mechanism automatically. Automatic text evaluation for the coherence of the generated text is very important matter for NLG systems. Although, there are many researches that investigate the problem of an automatic text evaluation [5], few NLG researches have investigated this problem in their systems/models [6].

In this paper we propose a new model to generate an English text from semantic graph. The semantic graph representation called "Rich Semantic Graph" (RSG). RSG consists of set of classes' verb and noun objects that have attributes and behavior. This model accesses the WordNet ontology to generate multiple texts according to the word synonyms. In addition, the model enables users to determine the output text style by selecting one of two writing styles (Cause/Effect and Description/Narration). Furthermore, in our model, the generated multiple texts are evaluated and ranked based on two criteria: most frequently used words and discourse sentence relations. The advantage of this model can be exploited in any application based on the input RSG. If the accepted RSG represents a reduced graph for bigger one corresponding to a document, the model will generate a summary for that document. In the same way, if the model accepted corresponding graph represented in some language, this graph correspond to a graph in another language it can be used in machine translation. Our model will consist of five tasks: Text planning, sentence planning, surface realization, writing styles and evaluation the first three tasks will be divided to subtasks.

3 NLG MODLE CONCEPTUAL VIEW

The process to generate text can be as simple as keeping a list of canned text that is copied and pasted, possibly linked with some glue text. The results may be satisfactory in simple domains such as horoscope machines or generators of personalized business letters. However, a sophisticated NLG system needs to include processes of planning and merging of information to enable the generation of text that looks natural and does not become repetitive.

Most NLG systems use ontology as a knowledge source for generating the final text from the input representation. Domain ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain [7]. WordNet is considered an example of ontology to English language. WordNet is an online lexical reference system in which English nouns, verbs, adjectives and adverbs are organized into synonym sets or *synsets* [8]. In this paper, WordNet will be exploited in the sentence planner and in the evaluation.

As shown in figure1, the proposed NLG model takes a semantic representation in the form of rich semantic graph (RSG) and generates multiple texts. This semantic graph contains the information needed to generate the final texts. To achieve its task, the model accesses the domain ontology which contains the information needed in the same domain of RSG to generate the final texts. Also, the model exploits the WordNet ontology to retrieve the word synonyms, and hence the model outputs multiple texts.

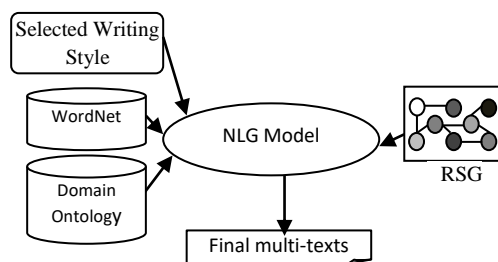


Figure 1: NLG Model Conceptual View

In general, there are three typical phases composing the NLG system. Firstly, the text planner phase that aims to select the appropriate content material to be expressed in the final text. Secondly, the sentence planner phase that specifies the sentence boundaries, and generates and orders an intermediate paragraphs. Finally, the sentence realization phase that generates corrected paragraphs grammatically. In addition to these phases, the proposed model includes the writing style selection phase to help the user to choose the style of writing for the output text. Because of generating multiple texts, the phase of text evaluation is proposed in our model to evaluate the final multiple texts based on the most frequently used words using WordNet ontology and the relations between sentences.

4 NLG MODEL ARCHITECTURE

The detailed architecture of our model is shown in figure 2. The model architecture contains five phases namely: Text planner, Sentence planner, Surface realization, Writing style selection, Evaluation. Each one of these phases may include more than one process. The Text planner includes content determination process the entire objects needed for the generations of the text are selected in this process. The Sentence planner includes four processes lexicalization, discourse structure, aggregation and referring expression. In the lexicalization process all the noun and verb word synonyms are considered, ranked and used after that in the generation of multiple texts. The discourse structure process include generate, order and relate simple sentences. The

Aggregation process combines the simple sentences and generates a simple paragraph. Finally, the referring expression process it involves selecting a pronoun or phrase that will identify an entity in the current context. Accepting from the sentence planner a list of sentence specifications, the sentence realizer's objectives are to determine the grammatically correct order of sentences. Inflect words for tense, number, and so on, as required by the language. Add punctuation, capitalization. After choosing the words and generating the paragraphs. The selection of the paragraph which matches with the given style of writing will be done using Essay writing styles selection. Finally Text evaluation, it is very important to see whether the text is coherent, good written and easy to read or not. NLG model are going to evaluate the coherence of the generated text. Text coherence evaluation is used for assessing whether parts of the document coherent or not. In the following points, each of these phases is discussed in details.

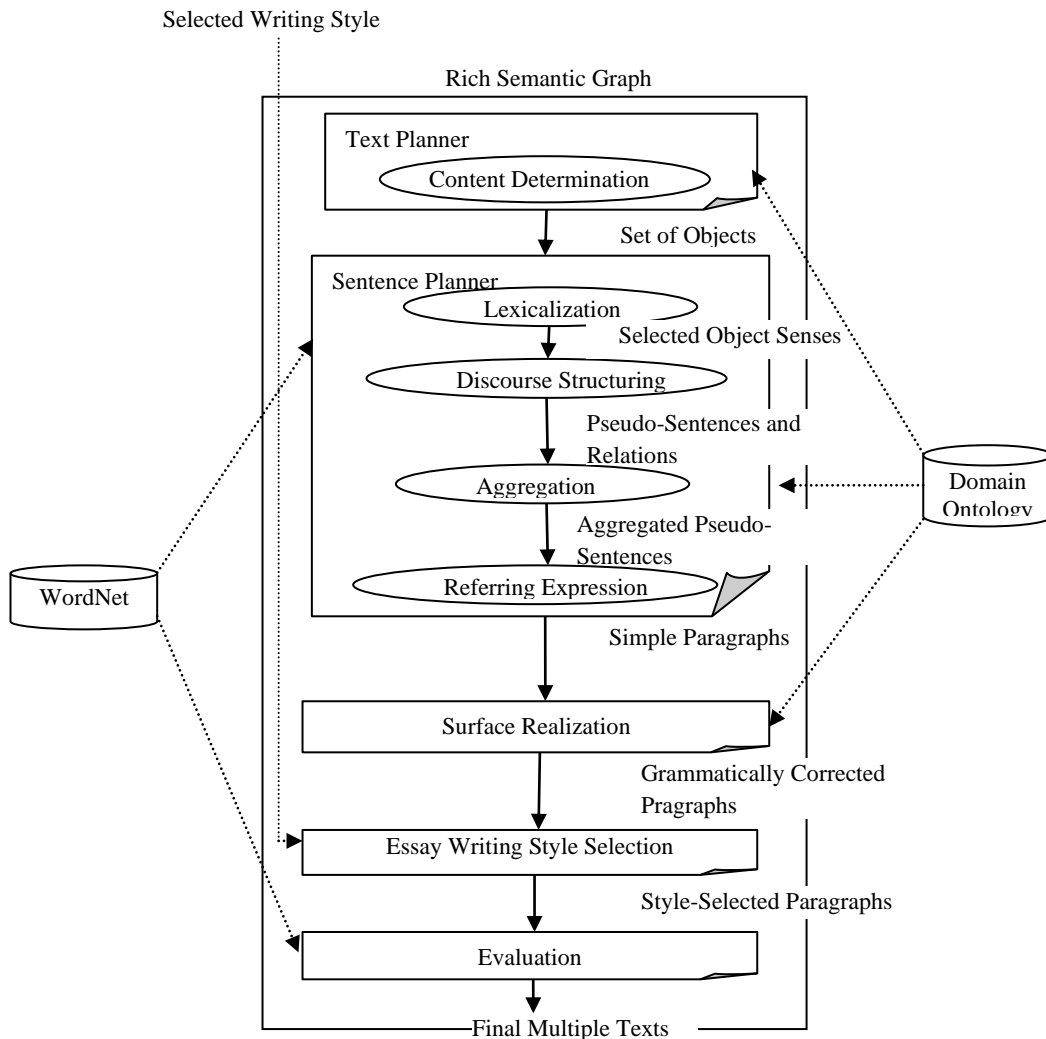


Figure 2: NLG Model Architecture

A. The Text planner phase

This phase includes one process called “**process content determination**”. It involves deciding what information should be included in the generated text and extracting this information from the knowledge base of an application. In our model, to preserve all semantic information embedded in the input semantic representation (RSG), all graph objects are considered to be passed to the sentence planner phase.

B. The Sentence Planner phase

The main objective of the second phase is to improve the fluency or understandability of the text. For example, the words of the text should be appropriate to both the user and context, the clauses should exhibit no unintentional redundancy, and similar sentences with the same subject should be aggregated. Sentence planner is one of the most important phases in NLG systems. In this model, the sentence planner receives noun and verb objects and generates simple paragraphs. To achieve its objective, the sentence planner consists of four main processes:

1) *Lexicalization process*: The first process in sentence planner is Lexicalization. Its objective is choosing a particular verb or noun object synonyms that are required to achieve the specified content by accessing the WordNet ontology. In this process, all the synonyms of the noun and verb objects in the input semantic graph are considered. To select the most appropriate synonyms, a weight (W) is assigned for every synonym. This weight is calculated based on equation no. 1. Where F is the word usage frequency and Si is the semantic information.

$$W = (F + Si)/2 \quad (1)$$

These equations are applied on the synonyms of all the selected noun and verb objects. The weight generated from applying the equation 1 on every synonym will have a value from 10 to 0. The highest weight will be given to the most appropriate word. An experimental test is made and we have found that the best threshold value is 8. The words with this value are selected and out of this weighted words. The words with weight from 8 to 10 will be selected.

2) *Discourse Structuring*: It is the process of building a structure that contains the object selected in lexicalization and generating pseudo-sentences which are the first form of the generated sentences. In this model, the discourse relations will be taken as input. The model uses the PDTB relations (Pann Discourse Tree Bank Model) [9]. The discourse role of the object is defined in the input Semantic graph. As the discourse relation type plus the argument span in which the object is located in the input semantic graph. Generates relations between the pseudo-sentences and connects these pseudo-sentences with each other using the relation given to the model. The details of this process are very application dependent [10].

In this process, algorithm is shown in figure 3 is used. In the generation of pseudo-sentences is started first with the nouns which have the largest number of attributes. Then apply the algorithm on all the nouns and verbs for all the given objects.

For the given semantic objects:
 1-For all the nouns
 Sort all the objects descending based of the number of attributes
 Compose a pseudo-sentence for every attribute as follow
 Form < Object Name Attribute > is < Object name > or
 Form < Object Name Attribute > < Attribute name > is <attribute value>
 Apply for all attributes except name attribute.
 2- For all verbs related to the above nouns
 Compose a Pseudo-Sentence for every attribute
 < Object subject > < Verb Object > <Attribute values>

Figure 3: Discourse Structuring Algorithm

After using the discourse structuring steps in the generation of the pseudo-sentences. In the second part in the discourse structuring process pseudo-sentences will be related with semantic relations. The discourse relations of a text will be written with the input semantic graph. It will use each PDTB explicit/implicit relation with two levels of relation types.

3) *Aggregation*: It is deciding how pseudo-sentences should be combined into simple paragraph. Aggregation is done by combining multiple pseudo-sentences into one single paragraph. In this process the following steps will be applied: Grouping and collapsing[11].Grouping and collapsing are divided into subject grouping and predicate grouping. Subject Grouping, the principal aggregation operation, implies collecting clauses with common elements [11]. Figure 4 shows an example for the process of subject grouping.

Formula: Subject₁ phrase₂ + Subject₁ phrase₃... Subject₁ phrase_n => Subject₁ (phrase₂ + Phrase₃... phrase_n)

a) Sally is student.
 b) Sally is in final year.
 c) Sally is going to graduate this semester
 d) Sally did not see Sarah since last year

Sally (is student in final year, going to graduate this semester, did not see Sarah since last year)

Figure 4: Example of subject grouping

In predicate grouping two or more propositions with identical predicates are aggregated to form a single proposition with a compound subject. Figure 5 shows an example for the process of predicate grouping.

Formula: Phrase ₂ Predicate ₁ + Phrase ₃ Predicate ₁ ... Phrase _n Predicate ₄ => (phrase ₂ + Phrase ₃ ... phrase _n) Phrase
a) Sally is student. b) Sarah is student.
Sally and Sarah are students.

Figure 5: Example of Predicate grouping

In the aggregation process multiple simple paragraphs will be generated. Threshold value is generated to select part of these simple paragraphs.

4) *Referring expression*: It involves selecting a pronoun or phrase that will identify an entity in the current context. Generating referring expressions in open domains algorithm will be modified to be used in the model [12]. The algorithm in figure 6 contains two main parts. The first part is for nominal's (head a noun phrase). The second is for the verbs and for subject in the simple sentence.

<p>FOR each nominal in aggregated Pseudo-Sentences SS DO</p> <ol style="list-style-type: none"> 1. IF a nominal is similar to the head noun of the object of any aggregated pseudo-sentences THEN <ol style="list-style-type: none"> (a) $SQ = SQ + 4$ (b) Fatten that relation for Pseudo-Sentences SS, i.e., add the attributes of the object of the relation to the attribute list for pseudo-sentence. (c) Replace every pseudo-sentence S which has $SQ > 4$ with the proper pronoun: Restrict the replacement of the nominal by only three pseudo-sentences then repeat the nominal again. 2. for each word or verb except nominal's w_i IF w_i is similar to the head of SS THEN Add all attributes of w_i to the attribute list and calculate their DQs. 3. Calculate DQ for the relation 4. If there is any repeated w_i in the case of words THEN delete the repeated word.
--

Figure 6: Referring Expression generation Algorithm

C. Surface Realization phase

The third phase is surface realization involves Accepting from the sentence planner a list of sentence specifications, the sentence realizer objective is to determine the grammatically correct order of words. Inflect words for tense, number, and so on, as required by the language. Add punctuation, capitalization. These tasks are language-dependent.

Simplenlg Simple natural language generation will be used in the model[13]. It can be used to help write a paragraph which generates grammatically correct English sentences. In this phase we are going to take the selected paragraph, specify the required input to the *simplenlg* system by writing a simple program.

D. Essay writing styles selection

After choosing the words and generating the paragraphs. In this phase the selection of the paragraph which matches with the given style of writing will be done. Based on the style of writing the output wanted to be. There are eight popular ways to structure essays: Description, Narration, Comparison Contrast, Process, Classification, Division, Cause and Effect, Exposition, Argumentation, Persuasion and Definition. In our model we are going to focus on two ways: description and cause and effect.

<p>For each paragraph:</p> <ul style="list-style-type: none"> Read the paragraph statement by statement Input the style wanted Repeat <ul style="list-style-type: none"> For each statement <ul style="list-style-type: none"> Search about the words refers to style Replace it by the word of the style wanted Reorder the sentences and the semantic relations Output the set of statement written in the style Aggregate the paragraph again with the modified style

Figure 7: Essay writing Algorithm

In this phase two style of writing will be focused on. The words to express the style of writing will be included. The words in every simple paragraph generated in the aggregation task will be compared with the words included for every style of writing. Then the model decides the paragraph is more compatible with which style of writing. After that the model output the paragraph with the style of writing wanted. Figure 7 shows the Writing style selection algorithm.

E. Evaluation

Text coherence evaluation is used for assessing whether parts of the document coherent or not. Evaluating topic coherence is a component of the larger question of what are good topics. What characteristics of a document collection make it more amenable to topic modeling[5]. Figure 8 shows the evaluation algorithm.

For all the sentence in the paragraph
 Create a grid where the rows represent the sentences and columns represent the objects
 Put the discourse role of every object in the sentences in the grid cells
 Calculate the grid cells
 Calculate the discourse role of every object in all the sentences multiply it by 2 the arguments of the relation
 Calculate all the roles for all the objects r_i
 Compare the value r_i with the values of all the grid cells and give a number form ten

Figure 8: Evaluation Algorithm

After applying the evaluation algorithm on the generated simple paragraphs the model will rank the evaluated paragraph and give every paragraph a value from ten to five based on the noun and verbs that most frequently used.

5 EXAMPLE

NLG model tasks are going to be applied on a real example. In this example the text planning is going to generate sentence plan. The details in this graph are added to our ontology. The ontology contains the classes, the individuals and attributes or properties which generated from this graph. The input to text planning is rich semantic graph in figure 9.

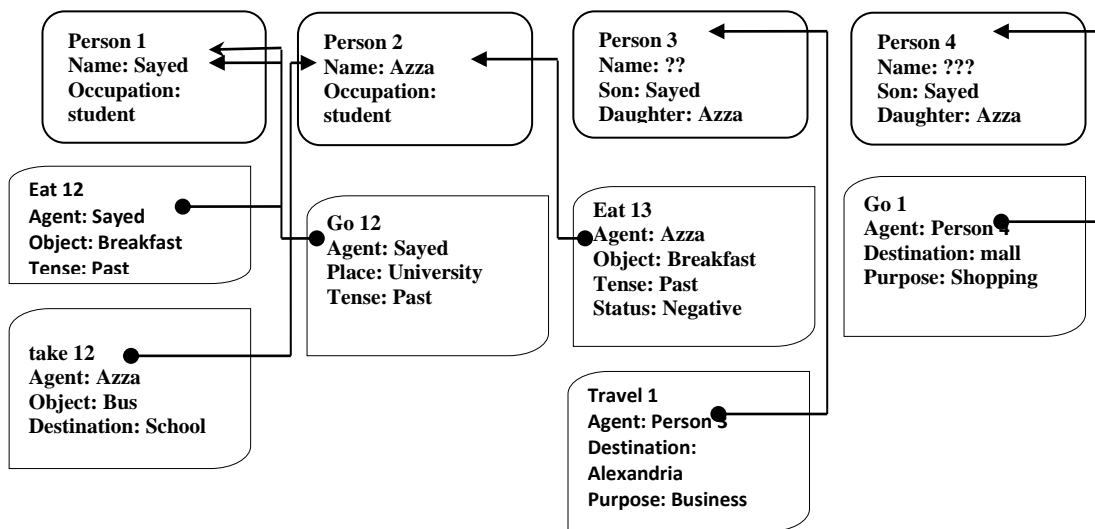


Figure 9: The Input Semantic Graph

The input in the NLG model is a semantic graph shown in figure 9. The output is a set of evaluated simple paragraphs shown figure 10.

A. Text planner

Content determination: In this task the objects which are going to use from the rich semantic graph will be illustrated. In this example we are going to select all the objects in the rich semantic graph.

B. Sentence planning

1) *Lexicalization*: In this example, the selected noun and verb objects will be taken to find there synonyms using WordNet. The verb “eat” has 17 synonyms. By applying the equation (1) a weight is assigned to these synonyms. Using the threshold eight, two synonyms are selected. The verb “go” has 100 synonyms. By applying the equation (1) a weight is assigned to these synonyms. Using the threshold eight, four synonyms are selected. The verb “travel” has 14 synonyms. By applying the equation (1) a weight is assigned to these synonyms. Using the threshold eight, four synonyms are selected. The verb “take” has 100 synonyms. By applying the equation (1) a weight is assigned to these synonyms. Using the threshold eight, 12 synonyms are selected. The noun “person” has three synonyms. By applying the equation (1) a weight is assigned to these synonyms. Using the threshold eight, four synonyms are selected. Figure 11 shows the output from the lexicalization phase.

1. Sayed is person. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(9)
2. Sayed is person. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(8)
3. Sayed is individual. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother locomotes to mall for shopping. he goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(7)
4. Sayed is individual. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(7)
5. Sayed is individual. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(7)
6. Sayed is soul. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is soul. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(6)

Figure 10: Final Output

2) *Discourse structuring*: In the discourse structuring we are going to apply the algorithm in figure 3. By following the algorithm steps based on our semantic graph in figure 11 and use the synonym of words selected in the lexicalization. The following pseudo-sentences shown in figure 12 will be generated. In the model the first three generated pseudo-sentences based on the appearance in WordNet rank will be selected.

<p>The selected synonyms for noun Person: "individual, someone, somebody, mortal, soul with weight (9)</p> <p>The selected synonyms for verb eat: "feed, eat." the selected synonyms with weight (9.1)</p> <p>The selected synonyms for verb go: "travel, move, locomote." the selected synonyms with weight (8.2)</p> <p>The selected synonyms for verb travel: "go: travel, move, locomote." the selected synonyms with weight (8)</p> <p>The selected synonyms for verb take: "occupy, use up, lead, direct, conduct, guide, get hold of, assume, acquire, adopt, take on, read ." the selected synonyms with weight (9.5)</p>

Figure 11: The output form lexicalization

The second part of discourse structuring is to use semantic relations to link between these pseudo sentences. The PDTB (penn discourse tree bank) [9]. It is lexically-grounded annotations of discourse relations. The semantic relations: "explicit" and "implicit" are going to be used. The PDTB is going to be used to do the second part of discourse structure. Then start the aggregation using these simple rhetorical relations. The relations between the pseudo-sentences are implicit contingency and implicit expansion.

3) *Aggregation*: In the aggregation task the subject grouping is applied. The sentences in figure 12 will be generated. The output from subject grouping for the subject Sayed are (54) simple paragraph. The outputs from subject grouping for the subject Azza are (18) simple paragraph. We are going to select the first six aggregated pseudo-sentences from each group. The results from aggregating the selected set of groups from the aggregated pseudo-sentences are shown in figure 13. The output from grouping and aggregation are (36) sentence. The first six simple paragraphs are going to be selected the given aggregated Pseudo-sentences will be selected so the output of the aggregation will be as figure 13.

The relations between the pseudo-sentences are implicit contingency and implicit expansion.

4) *Referring expression*: It identifies the intended referent(s). Algorithm in figure 5 will be used. Figure 14 shows the output of the sentence planning. Six possible several simple paragraphs are shown. The next phases of natural language generation system will use the simple paragraphs generated from sentence planning.

Person1	
	<p>Sayed is person Sayed is individual Sayed is soul</p>

	Sayed occupation is student Sayed sister is Azza
Person2	
	Azza is person Azza is individual Azza is soul
	Azza occupation is student Azza brother is Sayed
Person3	
	Father of Sayed is person Father of Azza is person
Person4	
	Mother of Sayed is person Mother of Azza is person
Eat12	
	Sayed eat breakfast Sayed feed breakfast
Go12	
	Sayed go to university Sayed travel to university Sayed locomote to university
Take12	
	Azza take the bus to school Azza occupy the bus to school Azza use up the bus to school
Eat13	
	Azza did not eat breakfast Azza did not feed breakfast
Go1	
	Sayed mother go to mall for shopping Sayed mother travel to mall for shopping Sayed mother locomote to mall for shopping
Travel1	
	Sayed father travel to Alexandria for business Sayed father locomote to Alexandria for business Sayed father go to Alexandria for business

Figure 12: The generated pseudo-sentences

1. Sayed is person, occupation is student , sister is Azza, father travel to Alexandria for business, mother go to mall for shopping, go to university, eat breakfast ,Azza is person, occupation is student, brother is Sayed, take bus to school, did not eat breakfast
2. Sayed is person, occupation is student , sister is Azza, father go to Alexandria for business, mother go to mall for shopping, go to university, eat breakfast ,Azza is person, occupation is student, brother is Sayed, take bus to school, did not eat breakfast
3. Sayed is individual, occupation is student , sister is Azza, father locomote to Alexandria for business, mother go to mall for shopping, go to university, eat breakfast ,Azza is person, occupation is student, brother is Sayed, take bus to school, did not eat breakfast
4. Sayed is individual, occupation is student , sister is Azza, father travel to Alexandria for business, mother locomote to mall for shopping, go to university, eat breakfast ,Azza is individual, occupation is student, brother is Sayed, take bus to school, did not eat breakfast
5. Sayed is individual , occupation is student , sister is Azza, father go to Alexandria for business, mother locomote to mall for shopping, go to university, eat breakfast ,Azza is individual, occupation is student, brother is Sayed, take bus to school, did not eat breakfast
6. Sayed is soul, occupation is student ,sister is Azza, father locomote to Alexandria for business, mother locomote to mall for shopping, go to university, eat breakfast ,Azza is soul, occupation is student, brother is sayed, take bus to school, did not eat breakfast

Figure 13: The selected generated grouping for the simple paragraphs

1. Sayed is person, his occupation is student , her sister is Azza, his father travel to Alexandria for business, Sayed mother go to mall for shopping, he go to university, he eat breakfast, Azza is person, her occupation is student, her brother is sayed, azza take bus to school, she did not eat breakfast.
2. Sayed is person, his occupation is student , her sister is Azza, his father go to Alexandria for business, Sayed mother go to mall for shopping, he go to university, he eat breakfast, Azza is person, her occupation is student, her brother is sayed, azza take bus to school, she did not eat breakfast.

3. Sayed is individual, his occupation is student , her sister is Azza, his father locomote to Alexandria for business, Sayed mother go to mall for shopping, he go to university, he eat breakfast, Azza is person, her occupation is student, her brother is sayed, azza take bus to school, she did not eat breakfast.
4. Sayed is individual, his occupation is student , her sister is Azza, his father travel to Alexandria for business, Sayed mother locomote to mall for shopping, he go to university, he eat breakfast, Azza is individual, her occupation is student, her brother is sayed, azza take bus to school, she did not eat breakfast.
5. Sayed is individual , his occupation is student , her sister is Azza, his father go to Alexandria for business, Sayed mother locomote to mall for shopping, he go to university, he eat breakfast, Azza is individual, her occupation is student, her brother is sayed, azza take bus to school, she did not eat breakfast.
6. Sayed is soul, his occupation is student , her sister is Azza, his father locomote to Alexandria for business, Sayed mother locomote to mall for shopping, he go to university, he eat breakfast, Azza is soul, her occupation is student, her brother is sayed, azza take bus to school, she did not eat breakfast.

Figure 14: The generated simple paragraphs with referring expression

C. Surface realization

In this phase simplenlg Simple natural language generation will be used[13]. It can be used to help write a program that generates grammatically correct English sentences. It's a library (not an application), written in Java, which performs simple and useful tasks that are necessary for natural language generation. Because it's a library, it will be needed to write our own Java program which makes use of *simplenlg* classes. The output from the surface realization phase is shown in figure 15.

1. Sayed is person. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
2. Sayed is person. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
3. Sayed is individual. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
4. Sayed is individual. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
5. Sayed is individual. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother locomotes to mall for shopping. he goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
6. Sayed is soul. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is soul. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.

Figure 15: The grammaticality correct simple paragraphs

D. Essay writing styles selection

In this phase the input given to the task will the wanted style of writing and the set of generated paragraphs. Then the appropriate paragraph will be chosen based on the selected one. If the entered the Description / Narration writing style then the output would be the generated text in figure17. If the entered style is cause / effect the output would be as shown in figure 16.

1. Sayed is individual. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother locomotes to mall for shopping. he goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
2. Sayed is person. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
3. Sayed is person. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.
4. Sayed is individual. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.

Figure 16: The simple paragraphs with the input description/ narration style

E. Evaluation

We are going to use Automatic Evaluation of Text Coherence using discourse: Models and Representations. Figure 18 show the output of the evaluation phase for the description / narration style. Figure 19 show the output of the evaluation for the cause /effect style. And use the WordNet rank for the most frequently used words.

1. Sayed is individual. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.

- | |
|---|
| 2. Sayed is soul. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is soul. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast. |
|---|

Figure 17: The simple paragraphs with the input cause / effect style

1. Sayed is person. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(9)
2. Sayed is person. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(8)
3. Sayed is individual. His occupation is student. Her sister is Azza. His father goes to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(7)
4. Sayed is individual. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother goes to mall for shopping. He goes to university. He ate breakfast. Azza is person. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(7)

Figure 18: The evaluated simple paragraphs

1. Sayed is individual. His occupation is student. Her sister is Azza. His father travels to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is individual. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(7)
2. Sayed is soul. His occupation is student. Her sister is Azza. His father locomotes to Alexandria for business. Sayed mother locomotes to mall for shopping. He goes to university. He ate breakfast. Azza is soul. Her occupation is student. Her brother is Sayed. Azza takes bus to school. She did not eat breakfast.	(6)

Figure 19: The evaluated simple paragraphs

6 CONCLUSION AND FUTURE WORK

This paper presents the NLG model, which takes set of noun and verb objects as an input and generates simple possible paragraphs. The model composed of five tasks Text planning, sentence planning, surface realization, writing styles and evaluation. The first task will generate the selected noun and verb synonyms which will be used. The sentence planning task will generate the simple paragraphs which is not grammatically correct. In the surface realization task, the generated simple paragraph will be grammatically corrected. The writing styles will take an input the selected writing style and output the simple paragraph with this style in our model there is two writing style descriptive and cause and effect. Finally evaluation, the coherence of the generated simple paragraph is going to be evaluated. An example of the NLG model will be discussed to illustrate how the model works.

REFERENCES

- [1] Douglas. E. Appelt. "Planning English Sentences". Cambridge University Press, 1985.
- [2] Dimitrios Galanis, George Karakatsiotis, Gerasimos Lampouras, and Ion Androutsopoulos. "An open-source natural language generator for OWL ontologies and its use in prot'eg'e, and second life". In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session, pages 17–20, Morristown, NJ, USA. Association for Computational Linguistics, 2009.
- [3] Niels Schutte, "Generating Natural Language Descriptions of Ontology Concepts" Dublin Institute of Technology Dublin, Ireland, 2009.
- [4] Mostafa Aref and Srinivasa Desiraju, "An Object-Oriented Approach for Discourse Analysis," Proceedings of the 15th International Conference on Computer Applications in Industry and Engineering, San Diego, California, PP 7-11, 2003.
- [5] Ziheng Lin, Hwee Tou Ng and Min-Yen Kan. "Automatically Evaluating Text Coherence Using Discourse Relations". In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011), Portland, Oregon, USA, June. (2011).
- [6] Amanda M. Holland-Minkley, Regina Barzilay, and Robert Constable. Verbalization of high-level formal proofs. In Proceedings of AAAI'99.
- [7] Gruber, Thomas R. "A translation approach to portable ontology specifications" (PDF). Knowledge Acquisition 5 (2): 199–220. (June 1993).
- [8] Bellare, K., Sarma, A.D., Sarma, A.D., Loiwal, N., Mehta, V., Ramakrishnan, G., Bhattacharya, P., "Generic text summarization using WordNet. In: Proc. Internat. Conf. on Language Resources and Evaluation". 2004.
- [9] Webber. "The Penn Discourse Treebank 2.0". In Proceedings of the 6th Int. Conference on Language Resources and Evaluation (LREC 2008). 2008.
- [10] Reiter and R. Dale. "Building natural language generation systems". Cambridge, U.K. Cambridge University Press, 2000.
- [11] Hercules Dalianis, Eduard Hovy "Aggregation in Natural Language Generation" EWNLG-93, Proceedings of the 4th European Workshop on Natural Language Generation, Pisa, Italy, 1993.
- [12] A. Siddharthan and A. Copestake. "Generating referring expressions in open domains". In Proceedings of ACL 2004.
- [13] Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009), pages 90–93, Athens, Greece, March. Association for Computational Linguistics.
- [14] Robert Stevens, Carole A. Goble and Sean Bechhofe, "Ontology-based Knowledge Representation for Bioinformatics ", Department of Computer Science and School of Biological Sciences University of Manchester Oxford Road, 2000.
- [15] Cole R.A., Joseph, Mariani, & Hans, Uszkoreits. "Survey of the State of the Art in Human Language Technology "uk: Cambridge University Press and Giardini, 1997.

English to Arabic Statistical Machine Translation Employing Pre-processing and Morphology Analysis

Shady Abdel Ghaffar¹, Mohamed Waleed Fakhr²

¹*Faculty of computing and Information Technology, Arab Academy for Science and Technology*

Sheraton, Cairo, Egypt

¹shady_fcis@yahoo.com

²*Faculty of Engineering, Electrical Engineering Department, University of Bahrain*

Eissa Town, Bahrain

²mfakhr@uob.edu.bh

Abstract— In this paper we show how to achieve a significant increase in Bleu score in case of English to Arabic Statistical Machine Translation (SMT) by making some preprocessing for both English and Arabic and also using Morphological splitting of Arabic. The preprocessing involves numbers, dates and person names clustering. The morphological splitting uses Columbia University Arabic Morphological analysis tool (MADA) and the SMT is using MOSES and GIZA++ tools.

1 INTRODUCTION

Machine Translation (MT) is the use of computers to automate some or all of the process of translating from one language to another. Many useful applications for MT including Cross-Language Information Retrieval (CLIR) which is a type of information retrieval where the language of the query and the language of the searched text are different; for example, searching Arabic text using English query. The World Wide Web nowadays contains tons of useful information presented in many languages. A typical internet user needs a machine translation system that is capable of delivering ideas and concepts presented in other languages to the user's language. Translating weather forecasting, News and computer manuals are very popular applications for MT. One-to-Many MT is applicable in translating manuals, books, and news. Many-to-one translation is required in translating the web content. An example for Many-to-many translation is the European Union where 23 official languages need to be inter-translated. Machine translation is a hard problem for several reasons; first languages are different at several levels; we have typological differences. At word level, words in different languages have different number of morphemes varying from one morpheme per word like Vietnamese (isolating languages), to many morphemes per word (polysynthetic languages). At syntactic level we have SVO languages (Subject Verb Object languages) like French, English and German, SOV languages (Subject Object Verb languages) like Hindi and Japanese, and VSO languages (Verb Subject Object languages) like Arabic and Hebrew. In addition we have lexical divergence; a word may have multiple senses, but only one in the context so, we need to have word sense disambiguation. On the other hand a word might be translated using one or more words in the target language [1].

Arabic is a highly inflected language where each word is inflected for gender and number. In addition a word may construct a meaningful sentence in its own. This makes word level alignment algorithms give bad alignment results [2]. For this reason we need to think of a way to improve the alignment quality to achieve good translation results. We can make use of morphological analysis as a preprocessing to resolve word level ambiguity and generate good alignment.

In this paper we discuss various preprocessing techniques that affect the Bleu score for English to Arabic statistical machine translation in addition we show that using morphology analysis enhances the Bleu score. Section II describes various machine translation techniques. Section III describes related works for both English to Arabic and Arabic to English SMT. In section IV we discuss several preprocessing tasks that affect the Bleu score when translating from English to Arabic. Then section V describes applying morphology analysis. Section VI describes post processing. Then section VII describes the baseline experiment and how the preprocessing affects the Bleu score. Finally MADA splitting experiments and how we make use of Morphology analysis. Section VIII is the discussions and conclusions and finally section IX is the future work.

2 MT APPROACHES

The different MT approaches can be grouped into two main camps, the rule based (RBMT) and the statistical based (SMT) approaches [1, 3].

RBMT approaches based on explicit rules those are put by expert linguists. In its pure form RBMT can be applied at different levels including Syntactic Transfer which uses hard coded rules to figure out the syntactic mapping between the source and the target language, other technique is the Interlingua MT, which attempts to model semantics. In general RBMT requires rules and

dictionaries which models the mapping between the source and the target language at the lexical and the syntactic levels those rules are developed manually or semi-automatically by language experts and software developers.

SMT is corpus based. SMT make use of translation samples called parallel/bilingual corpus. SMT in its basic form do the following. Given a sufficient sample of parallel text that is human translated the words are automatically aligned for each sentence pair. Then a translation model is learnt from the word alignment. The translation model basically models the words sequence mapping between the source and the target language. Then a decoder combines the translation model together with a language model for the target language to generate a ranked list of optimal translations.

RBMT was dominating the field of MT for many years; however over the last two decades researches for SMT have become very successful. The main motivation for this is the explicit linguistic rules can be probabilistic and can be learnt from parallel corpora. The last few years have witnessed an increasing interest in hybrid approaches between SMT and RBMT these approaches make use of both linguistic rules and statistical techniques. The most successful of such attempts so far are solutions that build on statistical corpus-based approaches by strategically using linguistics constraints or features [3].

3 STATISTICAL MACHINE TRANSLATION

SMT make use of the Bayesian Noisy Channel model. For example in case we are translating from English to Arabic the model assumes that the Arabic sentence has been distorted by the noisy channel as a result we have the English sentence [1, 3]. Our task is to recover the original Arabic sentence. In other words we need to find the proper Arabic sentence that is the most probable translation for a given English sentence as shown below using the Bayes probability rules:

$$A^{\wedge} = \operatorname{argmax}_A P(A | E) = \operatorname{argmax}_A P(E|A) * P(A) \quad (1)$$

$P(A|E)$ represents the faithfulness of mapping between the source and target languages, while the $P(A)$ represents the fluency of the translated target language sentence.

The noisy channel model requires three components. Translation model, language model and a decoding algorithm to find the sentence that maximizes the above equation.

$P(E|A)$ is the translation probability (the probability that the given English sentence is mapped to the generated Arabic sentence). We can estimate it by multiplying phrase translation probability and the distortion probability (reordering probability). We can think of any other models that maximizes the translation probability.

We call phrase translation probabilities Phrase Table is a bilingual mapping between source and target phrases and the mapping probability. Phrase table is extracted from the word level alignment. A phrase is a group of contiguous words.

Many models have been developed to generate word alignment given large parallel copra including EM algorithm, IBM model 1, 2, 3 and HMM based word alignment [1, 3].

Decoding algorithm searches the phrase table for the set of phrases that translates a given sentence and maximizes equation (1). Best first search algorithms are used like A^* and beam-search algorithm.

4 RELATED WORK

Arabic is a highly inflected language. Words are inflected for gender, number and some grammatical cases, but English is not. This mismatch between English and Arabic makes automatic word alignment between sentences pairs is a non-trivial problem. Therefore, efforts have been made to make English phrases match Arabic phrases to improve automatic alignment quality. In prior work [2] it has been shown that morphological segmentation of Arabic source makes a significant increase in Bleu score in Arabic to English SMT. However, English to Arabic SMT requires recombination. The better the recombination is the higher Bleu score is achieved. English to Arabic SMT is more difficult than Arabic to English SMT since the output in this case is segmented Arabic which requires recombination to construct Arabic words. The Recombination problem is non-trivial problem because Arabic is highly inflected language. In prior work [4] several recombination techniques were introduced. Those techniques are recombination table and a set of hard coded morphological rules that are obtained from the training set. In this paper, we compare the word-based system with and without preprocessing with the splitting-based system with and without preprocessing.

5 PREPROCESSING

Before we do training for our machine translation system we have done some preprocessing to the parallel corpus. We do simple tokenization, removing punctuations, normalizing all forms of Alef Hamza to bare Alif and final ‘Y’ Alif Maksora to Yaa. Numbers, numeric dates, times and percentages are not translated and they are nothing, but noise that corrupts the automatic alignment and harms the language model. In addition there is a very large number of values for these categories and only few of them may appear in the training and tuning data. This decreases the quality of language model and alignment. As a preprocessing we replaced all numbers, numeric dates, times and percentages by special tags (B) for numbers, (C) for percentages and (Q) for dates. To improve alignment quality we choose the maximum sentence length to be 40 words.

Let us assume that we have a large Arabic corpus of 1 million Arabic words. Among these words we have say about 10,000 different numeric values. In this case if we build a language model without doing number normalization then each number in its own has a very small probability. What is more if we see a number which we have not seen in the training data the language model assigns a very small probability to this unseen token. On the other hand if we represent all numeric values as only one token say (B) the unigram probability for this token will be the sum of the unigram probabilities for all the different values presented in the training set. The advantage of this abstraction is when we see a numeric value in the test set which we have not seen in the training set the system assigns a considerable probability for this token as a result a higher probability for the sentence is obtained. If we tackle the problem from the alignment point of view. In the first case the number of the unique tokens is larger than the number of the unique tokens in case of numeric values normalization as a result the automatic alignment in the second case is easier than in the first case.

The same idea can be applied on all other forms of non-translated words. In another experiment we replaced all person names in both Arabic and English by the tag (PRN). We will show that this preprocessing affects the Alignment quality and the Bleu score in a positive way.

6 MORPHOLOGY ANALYSIS

Each Arabic word has multiple possible analyses. When a word appears in a sentence it has only one analysis. We used MADA (an SVM based morphological analyzer by Nizar Habash [5]) to select the correct sequence of analysis for each word. This step is important because choosing the wrong analysis results in wrong prefix, suffix segmentation. In MADA experiments we used the following splitting scheme.

S1: Decliticization by splitting off each conjunction clitic (w+, f+, b+, k+, l+), definite article (Al+) , pronominal clitics including possession pronoun (+p) and object pronoun (+O:) . Note that Plural and subject pronouns are not splitted. S1 is summarized by (w+ f+ b+ k+ l+ Al+ REST +P: +O :). For example wAwlAdh (‘and for his kids’) it would be (w+ l+ Awlad +h) according to S1. Table I shows some examples of splitting clitics by MADA.

TABLE I
EXAMPLES OF SPLITTED ARABIC

Arabic	Buckwalter	Splitted Arabic
القطن المصري يتفوق على القطن الأمريكي	AlqTn AlmSry ytfwq Ely AlAmryky	Al+ qTn Al+ mSry ytfwq Ely Al+ Amryky
10% زيادة على رسوم المزارات السياحية	10% zyAdp Ely rswm Al+ mzArAt Al+ syAHyp	(C) zyAdp Ely rswm Al+ mzArAt Al+ syAHyp
تعديل التعريف الجمركية على السلع المستوردة	tEdyl AltEryfp Aljmrkyp Ely AlsIE Almstwrpd	tEdyl Al+ tEryfp Al+ jmrkyp Ely Al+ sIE Al+ mstwrpd

This step is performed before doing automatic alignment using GIZA++. Splitting the Arabic words into it’s morphemes (Affixes and stem) helps GIZA++ to align Arabic affixes to its corresponding English words and enhances the alignment quality. The problem with non-splitted Arabic is there are many attached pronouns, but there are not in English (all pronouns are not attached) for example the Arabic word (سيقاتلوهم) (syqAtwhm) the corresponding English translation is (They will fight them). If we split (سيقاتلوهم) (syqAtwhm) to its morphemes it turns out to be (s+ yqAtwA +hm). The EM algorithm for word alignment learns that the affix (+hm) is aligned to the English word (them) and the prefix (s+) is aligned to the English word (will). This could not be learnt in case of the non-splitted Arabic. For non-splitted Arabic the best results we can get when we align the Arabic word to all English words in the sentence those are the translation for this Arabic word for example the Arabic word (سيقاتلوهم) (syqAtwhm) is aligned to the English phrase (They will fight them) so if we see the English word (them) in the

testing set we will not be able to give appropriate translation. In addition the splitting reduces the unique number of Arabic vocabulary making the alignment task quite easier and maximizes the likelihood of the word-to-word alignment. The following table II shows the difference in the alignment for both splitted and non-splitted Arabic.

TABLE II
DIFFERENCE IN ALIGNMENT FOR BOTH SPLITTED AND NON-SPLITTED ARABIC

Splitted Arabic – English Alignment
<p>interior ministry refuses to supply banks with data about faltering clients</p> <p>Al+ dAxlyp trfD tzwyd Al+ bnwk b+ byAnAt En Al+ mtEvryn ال + داخلية ترفض تزويد ال + بنوك ب + بيانات عن ال + متعثرين</p> <p># Sentence pair (14) source length 4 target length 5 alignment score : 1.2029e-07</p>
Non-splitted Arabic – English Alignment
<p>interior ministry refuses to supply banks with data about faltering clients</p> <p>AldAxlyp trfD tzwyd Albnwk bbyAnAt En AlmtEvryn ال + داخلية ترفض تزويد ال + بنوك ب + بيانات عن ال + متعثرين</p> <p># Sentence pair (10) source length 21 target length 35 alignment score : 2.97411e-68</p>

The above table shows the difference in GIZA++ alignment for both splitted and non-splitted Arabic. For example bbyAnAt is wrongly aligned to the English word supply and En is wrongly aligned to NULL word. On the other hand in the splitted Arabic alignment b+ is correctly aligned to with and byAnAt is correctly aligned to data. In addition the alignment score in case of splitted Arabic is higher than the non-splitted Arabic alignment.

7 RECOMBINATION

Although the splitting improves alignment quality as shown in the previous section, the resulting translation will be splitted. We need to have a mechanism to do recombination. The recombination is not a simple task. The recombination difficulties can be summarized in the following

- Different context: Linguistically if we have affixes and stem we can generate many words for example (اراء +ه) ArA+h) can be recombined to (ارأه ArA&h) or (ارأه ArA}h) depending on the context.
- Some letters may be inserted when we do recombination for example (لكن +ي lkn +y) may be recombined to (لكني lkny) or (لكنني lkny).
- Some letters may be eliminated when we do recombination. For example (خدعوا +نا xdEwA+nA) recombined to (خدعونا xdEwnA).
- Some letters are replaced by another when we do recombination. For example the final Yaa maksoora is replaced by Alef when it is attached to a suffix. For example (خطى +هم xTy+hm) is recombined to (خطاهم xTAhm).

We observed the training and the tuning data to extract deterministic rules with high precession and low recall. In addition a recombination table is extracted from the training and the tuning data. A language model for non-splitted Arabic is used to take the decision in case of contextual ambiguity. The recombination techniques have been addressed in a prior work [4].

The advantage of splitting is sparseness reduction on the other hand the recombination is difficult because more than one possible word can be generated from a given stem affixes collection depending on the case ending. We can rely on a word based language model to choose the best recombined word, however this technique require a very strong language model that is built from a huge Arabic text to cover all case endings. Some recombination rules are listed in table III below.

TABLE III
RECOMBINATION RULES

Rules	Example
Final Taa Marboota is replaced by Taa Maftooha when the word is attached to a suffix	(اجندة +ها Ajndp+hA) → (اجندتها AjndthA)
Final Yaa Maksoora is replaced by Alef when the word is attached to a suffix.	(و +تتبنى w+ttbny+h) → (وتتبناها wttbnAh)
Final Hamza (ء) is replaced by either (ؤ) or (ة) depending on the context. (a language model is used here)	(+الغاء AlgA'+h) → (الغائه AlgA}h) or (الغاؤه → AlgA&h)
The prefix l+ when it followed by a prefix Al+ when we do recombination it turned out to be (لل ll+)	(ل +ال +حرية l+ Al+ Horyp) → (للحرية llHoryp)

In order to evaluate our recombination system we tried to recombine the test set and calculate the percentage of the missed combined sentences. The recombination error is around 1 %.

8 EXPERIMENTS

We carried out two main experiments. The first is the baseline experiment which does not involve morphology analysis. The second experiment involves using morphological analyzers MADA. We used the Arabic sentences in the training set to construct a 7-gram modified Kneser-Ney language model for the baseline and MADA experiments. We used SRI toolkit for language modeling [6]. Then GIZA++ [7] is used to obtain word alignment. MOSES scripts [8] are used then to extract the phrase table from the word aligned sentences we choose the maximum phrase length to be 8 words in case of the baseline experiment and 15 words in case of MADA experiment. MOSES scripts have been used to evaluate parameters together with the tuning set. Parameters are language model weight; phrase table weight and reordering table weight are tuned to achieve the highest bleu score over the tuning set. Bleu score is calculated after translating the test set using the tuned model.

We used an LDC parallel corpus catalog number LDC2004T18 and ISBN 1-58563-310-0. This corpus contains Arabic news stories and their English translations LDC collected via Ummah Press Service from January 2001 to September 2004. It totals 8,439 story pairs, 68,685 sentence pairs, around 2M Arabic words and 2.5M English words. The corpus is aligned at sentence level.

The Arabic sentences have been used to develop the language model. 2000 sentences pairs have been selected randomly for Tuning and another 2000 sentences pairs for Testing. The rest are left for training. Training data has been filtered to include sentences whose length is between 1 and 40 words for better alignment by GIZA++. 40,000 sentences pairs have been used for training.

A. Baseline Experiment

In this experiment we just used a simple tokenization for both Arabic and English. We applied the normalizations described in the preprocessing section. We repeated the experiment with and without using the numeric normalization. We repeated the experiment with and without person names clustering. Since Arabic named entity recognizer is not available and its accuracy is not as the English named Entity recognizer. We used Stanford English Named Entity Recognizer (NER) [9] to tag all person names in English text in the training set. Then we used Google translate to translate these names from English to Arabic. The parallel named entity list is manually revised. Finally all person names in both Arabic and English text are replaced by tag (PRN).

B. MADA Experiment

Training and tuning Arabic sentences are analyzed using MADA. Prefixes and suffixes are split. Prefixes are marked by a trailing plus sign and suffixes are marked by a beginning plus sign. So each word split into prefixes, stem and suffixes separated by spaces. After phrase table is constructed we removed all phrase table entries whose target phrase either starts with a suffix or ends with a prefix. We repeated this experiment with and without this post processing.

A set of recombination rules is extracted from the training data. A recombination table is extracted from the training data. Rules and the recombination table are tested on the testing set.

9 CONCLUSION

A significant increase in Bleu score can be achieved by doing simple numeric and date normalization. This is because numbers increase sparseness and is considered as out of vocabulary. If we group all numbers in a single token (B) the language model quality increase as shown in table 4. In addition word alignment quality increases as a result a higher Bleu score is achieved.

In MADA experiment phrase table filtering increases the Bleu score because it forces the decoder to output compatible affixes/stems as a result a well formatted Arabic words are generated.

Person names clustering in the baseline experiment decreases language model perplexity and improved the alignment quality. Person names are transliterated and they are infinite. They increase the number of vocabulary. Grouping these names in a single token (PRN) achieves 2 points in Bleu score as shown in table III.

TABLE III
COMPARES THE BASELINE EXPERIMENTS AND THE MADA-BASED EXPERIMENTS

Experiment	LM Perplexity	Bleu score
Baseline with basic letters normalization and basic tokenization	303	19.1
Baseline (Numbers/Dates Normalization +basic letters normalization)	269	24.8
Baseline (Number/Dates Normalization + basic letters normalization) + person names clustering	136.2	26.5
MADA using S1 splitting scheme (Without phrase table filtering)	139.2	27.05
MADA using S1 splitting scheme (With phrase table filtering)	139.2	27.39

ACKNOWLEDGMENT

We would like to thank Dr. Nizar Habash for providing MADA.

REFERENCES

- [1] Daniel Jurafsky, James H. Martin, *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*, publishing House of Electronics Industry, Beijing, China. 2004
- [2] Nizar Habash, Fatiha Sadat. *Arabic Preprocessing Scheme for Statistical Machine Translation*. In Proc. of HLT 2006.
- [3] Nizar Y. Habash, *Introduction to Arabic Natural Language Processing* 2010.
- [4] Ibrahim Badr, Rabih Zbib, James Glass. *Segmentation for English-to-Arabic Statistical Machine Translation*, In Proceedings of ACL'08 2008.
- [5] <http://www1.ccls.columbia.edu/~cadim/MADA.html>
- [6] <http://www-speech.sri.com/projects/srilm/>
- [7] Franz Josef Och, Hermann Ney. "Improved Statistical Alignment Models". Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hong Kong. October 2000.
- [8] MOSES. *A Factored Phrase-based Beam-Search Decoder for Machine Translation*, 2007 <http://www.statmt.org/moses/>
- [9] Stanford Named Entity Recognizer (NER) <http://nlp.stanford.edu/software/CRF-NER.shtml>

UNL Editor: An Annotation tool for Semantic Analysis

Sameh Alansary^{*1}, Magdy Nagi^{**2}, Noha Adly^{**3}

**Phonetic and Linguistics Department, Faculty of Arts, University of Alexandria
ElShatby, Alexandria, Egypt*

**Bibliotheca Alexandrina, Alexandria, Egypt*

¹sameh.alansary@bibalex.org

***Computer and System Engineering Department, Faculty of Engineering
Alexandria University, Egypt*

²magdy.nagi@bibalex.org

³noha.adly@bibalex.org

Abstract— This paper presents the UNL Editor as a tool for semantic annotation; discussing and describing the tool in details. The paper regards the tool in two aspects, describing its linguistic framework; explaining the logic on which the UNL Editor is based upon. Then, it goes to explain how this logic is applied when carrying out the semantic annotation of the natural language texts through presenting step by step instruction for using the tool. Finally, it exhibits the different usage of such a tool. However, in order to control the size of the paper, this paper is not concerned with addressing different linguistic issues of annotating natural language tests, or the linguistic difficulties arising within the process; it is only limited to presenting linguistic capabilities of the tool to prove its efficiency in semantic annotation¹.

1 Introduction

In the recent years, semantic Annotation has become an increasingly important research topic being a fundamental element of many Natural Language Processing applications like information retrieval, query answering and information extraction. Semantic annotation is additional information in a document that identifies or defines the semantics of a part of that document. In other words, Semantic annotation is about attaching sense tags, names, attributes, comments, descriptions, etc. to a document or to a selected part in a text [1]. Consequently, helping to bridge the ambiguity of the natural language when expressing notions and their computational representation in a formal language; by telling a computer how data items are related and how these relations can be evaluated. Thus, opening the way to numerous applications.

With rapidly growing amount of on-line web documents, web users need to find, share, and combine information more easily; urging researchers to focus on the creation and dissemination of innovative Semantic Web technologies to facilitate automated processing [2]. The semantic web depends entirely on semantic annotation. Hence, it would only seem natural to find number of tools designed to perform full semantic annotation for natural language texts. However, this is not the case, the number of tools intended to perform semantic annotation is extremely limited [3]. There have been several attempts to create a tool for analyzing natural language texts semantically. Some of the most worth of noting applications are GATE [4], KIM [5], Melita [6]. Nevertheless, none of the tools are totally automatic. Furthermore, these systems perform annotation on words and terminologies to indentify real world objects and their relationship in the text. None of them provide annotation above word level. A brief overview of some of these tools:

GATE (General Architecture for Text Engineering) is an infrastructure for development software components based on Human Languages. The GATE system provide many functionalities among them, it provides the functionality to annotate textual documents both manually and automatically. GATE uses JAPE [7] pattern matching engine for rule based Named Entity Recognition. JAPE is ontologically aware which can map the Named Entity to ontology classes during recognition. In GATE, the task of textual annotation is just defined more domain specific rules in addition to already available basic rules.

KIM is another ontology base semantic annotation system that uses a special knowledge base (KIMO) which has been pre-populated with 200,000 entities. KIM uses GATE, SESAME and Lucene for many information extraction tasks. KIM also uses version of ANNIE for Named Entity Recognition. KIM has a feature of automatically adding new instances found in text to Ontology. It also performs disambiguation step because many instances can be added to different places in ontology.

¹ Different issues of annotating Arabic texts semantically will be found in: [1] S. Alansary, "Semantic Annotation of Arabic Texts: Issues and Implications" (forthcoming)

Melita provides the interface to semantically annotate the textual document using Adaptive Information Extraction technique. This technique reduces the burden of text annotation on user. It starts with manual annotation of text by user and as user keeps on annotating text the system learns the annotation process. Melita uses Amilcare [8] which runs in background learning how to reproduce the inserted annotation.

Considering that semantic annotation has become a comprehensive concept, number of attempts have been made in order to integrate linguistic approaches in the analysis of natural language corpus, some of the most representative results were the Propbank project [9], FrameNet project [10]. The Proposition Bank project (Propbank) focuses on the argument structure of verbs and adding a layer of predicate-argument information, or semantic role labels, to the syntactic structures of the Penn Treebank. It aims to provide a broad-coverage hand annotated corpus with semantic annotation, enabling the development of better domain-independent language understanding systems. The FrameNet was initially a lexicographic project, engaged in building a lexicon with uniquely detailed information on the syntax and semantics of Lexical Units. More recently, since 2004 FrameNet has also been annotating continuous texts for deep semantic annotation. The FrameNet approach is based on linguistics theory of frame semantics. However worthy these attempts were, they were all manually done; none of which was performed by tools. Thus, the need to provide a tool designed with the intention of performing semantic analysis became undeniably clear.

In the context of the UNL (The Universal Networking Language), a semantically based interlingua to break language barriers between human languages, the UNDL Foundation in co-operation with Bibliotheca Alexandrina has started an initiative for building a tool for semantic annotation called the UNL Editor; a visual editor designed with the intention of providing full semantic annotation, thus analyzing natural language texts and, generating UNL documents. This tool is based upon a comprehensive visualization of the entire process of the annotation. It is uniquely designed on linguistic background; adopting certain linguistic theories closely related to computational linguistics in terms of using unified super sets of semantic relations [11] thus overcoming the problem of conflicting and confusing names [12], and making use of renowned lexical recourses; WordNet [13]. Moreover, it provides a powerful visual interface for working with UNL data both in a textual and graphical mode with friendly interface creating an appropriate environment for navigating through the needed steps of providing the analysis; it offers a visualization of the analysis through graphs which aids the representation of the semantic network created with every sentence analyzed. Most importantly, the UNL Editor's output offers the much need training data for semantic annotation due to the fact that the relations and concepts used are clearly defined as well as standardized within the UNL Editor framework, in addition the output is presented in a text file that could be easily used. The UNL Editor exhibits enormous flexibility and opportunities in handling natural language text due to the fact that it is designed upon linguistic framework, minding the complexity and richness of natural language, thus enriching the tool with all different kinds of options in order to handle the natural language, and paving the way for other applications through its easy to be used output.

This paper is concerned with presenting and explaining the UNL Editor as a manual tool for semantic annotation. It is divided into four sections; section 2 exhibits the linguistic framework which the design of the UNL Editor adopts as its bases; indicating why it is designed as such and linguistic theories are been adopted, section 3 is a detailed explanation accompanied with screenshots illustrating how this application could be used, section 4 represents the different usages of the UNL Editor as a tool for semantic annotation. Finally, Section 5 concludes the paper.

2 Linguistic Framework

The UNL Editor provides a means enabling the analysis of the underlying semantic relations composing the Natural Language sentences. It is designed on linguistic bases . On a semantic assumption or rather on semantic theory stating that a deep semantic analysis for a natural language text requires two levels of semantics; lexical semantics and grammatical semantics [14].

A. *Lexical Semantics*

It is the study of how and what the words of a language denote. In other words, lexical semantics is meaning at word level [15]. In the UNL Editor, lexical semantics is expressed through creating the nodes, a process in which every word or rather every concept in the sentence to be analyzed is matched with its corresponding ID, meaning that a single node may contain more than one lexical item; a compound word, as long as it is representing a single concept. For example the term "Holy Quran" represents single concept, therefore it would be considered one node, having a single ID. The ID is a nine-digit string that is distinct number and assigned to each concept. The dictionary, from which the IDs are extracted, is based upon the WordNet 3.0; a lexical database for English Language, contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs). The WordNet is considered to be the most prominent and widely used lexical resource for researchers in computational linguistics, text analysis, and many related areas [16]. In order to make the process of selecting the appropriate ID easier and for

more clarification to the concept, the UNL Framework made use of the set of information the WordNet attach to each concept, these information consist of a distinct ID, an abstract meaning (the gloss), the "synset" which is a set of one or more synonyms that are interchangeable in some context without changing the truth value of the proposition in which they are embedded, the corresponding part of speech and in some cases examples are shown. The right half of the interface is dedicated for the lexical semantics through the search pane, in which there are three search options are offered by exhibiting three tabs, each tab is dedicated for a different kind of search [3.1]. One of which offers the possibility of uploading dictionaries in attempt of providing an integrated development environment for UNL.

B. Grammatical Semantics

It has to do with meaning at sentence level; grammatical semantics is the study which explores the relation between patterns of meaning and grammatical structure. It is based on the assumption that the syntactic structure of the sentences overlaps with its semantics [17]. In the UNL Editor, grammatical semantics is expressed in terms of a range of semantic relations, and a list of attributes. There has always been a problem with using semantic relations as there is no formal basis for defining the notion clearly, making determining what should be qualified as a semantic relation and what is not confusing. In order to overcome this problem, the UNL Editor has proposed a unified super set of the semantic relations. These relations are highly standardized as each relation is clearly defined in the UNL framework. Table 1 contains all the 45 semantic relation that the tool includes and they are a closed set of relations. Moreover, it is a directed graph meaning that every relation has to start from certain node in order to convey the correct meaning. Relations are used to describe the objectivity information of sentences. In the UNL, relations are normally regarded as representations of semantic cases or thematic roles (such as agent, object, instrument, etc.) between concepts. They are used in form of arcs connecting a node to another node in a UNL graph. They correspond to two-place semantic predicates holding between two concepts. Relations are represented as two or three-character lower-case strings. Since there are similarities between the semantic relations and syntactic relations in name and function, it may seem that the labels used for relations are different names for special grammatical functions. However, the intention is that the labels used denote specific ideas rather than grammatical structures, the conceptual relations used in UNL are much more abstract than the grammatical relations found in syntax. In general, relations are always used to describe semantic dependencies between syntactic constituents. For example, in a sentence like "John breaks the door", the syntactic subject of the sentence is "John" and semantically it would be regarded as the "agt", whereas in a sentence like "the sugar melts in tea" the lexical item "sugar" is the syntactic subject of the sentence but semantically it would be considered as an object "obj".

Table I illustrates the UNL Editor semantic relations; definition, description and example to each relation

TABLE I
SEMANTIC RELATIONS

RELATION	DEFINITION	DESCRIPTION	EXAMPLE
Agt	Agent	a thing which initiates an action	car runs
And	And	a conjunctive relation between concepts	John and Mary
Aoj	thing with attribute	a thing which is in a state or has an attribute	Leaf is red
Bas	Basis	a thing used as the basis(standard) for expressing degree	Ten is three more than seven
Ben	Beneficiary	a not directly related beneficiary or victim of an event or state	To give one's life for one's
Cag	co-agent	a thing not in focus which initiates an implicit event which is done in parallel	To walk with John
Cao	co-thing with attribute	a thing not in focus is in a state in parallel	be with you
Cau	Cause	the cause of a state	The cause of the accident...
Cnt	Content	an equivalent concept	The Internet: an amalgamation
Cob	affected co-thing	a thing which is directly affected by an implicit event done in parallel or an implicit state in parallel	dead with Mary
Con	Condition	a non-focused event or state which conditioned a focused event or state	if you are tired, we will go straight home
Coo	co-occurrence	a co-occurred event or state for a focused event or state	was crying while running
Dur	Duration	a period of time during an event occurs or a state exists	work nine hours (a day)

Equ	Synonym	Synonym	the deconverter (a language generator)
Fmt	Range	a range between two things	the alphabets from a to z
Frm	Origin	an origin of a thing	a visitor from Japan
Gol	goal/final state	the final state of object or the thing finally associated with object	the lights changed from green to red
Icl	Inclusion	Inclusion	a bird is a (kind of) animal
Ins	Instrument	the instrument to carry out an event	look at stars through a telescope
Int	Intersection	indicates all common instances to have with a partner concept	an intersection of tableware and cookware
Man	Manner	the way to carry out event or characteristics of a state	move quickly
Met	Method	a means to carry out an event	solve ... with dynamics
Mod	Modification	a thing which restrict a focused thing	the whole story
Nam	Name	a name of a thing	his son "Hikari"
Obj	affected thing	a thing in focus which is directly affected by an event or state	the table moved
Opl	affected place	a place in focus where an event affects	pat ... on shoulder
Or	Disjunction	disjunctive relation between two concepts	Will you stay or leave?
Per	proportion, rate or distribution	a basis or unit of proportion, rate or distribution	eight hours a day
Plc	Place	the place an event occurs or a state is true or a thing exists	cook ... in the kitchen
Plf	initial place	the place an event begins or a state becomes true	traveling from Tokyo
Plt	final place	the place an event ends or a state becomes false	to travel to Boston
Pof	part-of	a concept of which a focused thing is a part	the preamble of a document
Pos	possessor	the possessor of a thing	John's dog
Ptn	Partner	an indispensable non-focused initiator of an action	compete with John
Pur	purpose	the purpose or an objective of an agent of an event or a purpose of a thing which exist	come to see you
Qua	Quantity	a quantity of a thing or unit	Two cups of coffee
Rsn	Reason	a reason that an event or a state happens	They can start because Mary arrived
Scn	Scene	a virtual world where an event occurs or state is true or a thing exists	win a prize in a contest
Seq	Sequence	a prior event or state of a focused event or state	Look before you leap
Src	Source	the initial state of an object or thing initially associated with the object of an event	The lights changed from green to red
Tim	Time	the time an event occurs or a state is true	leave on Tuesday
Tmf	initial time	the time an event starts or a state becomes true	work from morning to [till] night
Tmt	final time	the time an event ends or a state becomes false	be full till tomorrow
To	Destination	a destination of a thing	a train for London
Via	intermediate place	an intermediate place or state of an event	go ... via New York

Other additional information are being presented through attributes, representing information conveyed by natural language grammatical categories (such as tense, mood, aspect, number, etc) [18]. In opposition to relations, attributes correspond to one-place predicates; attributes are intended to be used as annotations made to nodes or hypernodes of a UNL hypergraph. Moreover, they are also a closed set. The names of attributes are always expressed in lower case words or expressions. Attributes are also used to express the range of concepts such as the concept indicate generic type of concept and so forth. On the one hand, relations and concepts are used to describe the objectivity information of sentences. On the other hand, attributes modify concepts or semantic networks to indicate subjectivity information such as about how the speaker views these states-of-affairs and his attitudes toward them and to indicate the property of the concepts. This includes phenomena technically called “speech acts”, “propositional attitudes”, “truth values”, etc. They are used to express logical expressions in order to strengthen the expressibility of the UNL. Attributes are divided into the following groups:

- | | | |
|-----------------------|--------------------|-------------------|
| 1) Aspect | 8) manner | 15) register |
| 2) Degree | 9) modality | 16) reference |
| 3) document structure | 10) numerals | 17) social deixis |
| 4) emotions | 11) person | 18) specification |
| 5) figure of speech | 12) place | 19) tense |
| 6) gender | 13) polarity | 20) time |
| 7) lexical category | 14) quantification | 21) voice |

Attributes are mainly used to convey three different kinds of information. First, the information on the role of the node in the UNL graph, as in the case of '@entry', that indicates the main (starting) node of a UNL directed graph; secondly, The information conveyed by bound morphemes and closed classes, such as affixes (gender, number, tense, aspect, mood, voice, etc), determiners (articles and demonstratives), adpositions (prepositions, postpositions and circumpositions), conjunctions, auxiliary and quasi-auxiliary verbs (auxiliaries, modals, coverbs, preverbs) and degree adverbs (specifiers); thirdly, The information of the (external) context of the utterance, i.e., non-verbal elements of communication, such as prosody, sentence and text structure, politeness, schemes, social deixis and speech acts.

3 How to Use the UNL Editor?

This section will present step by step instruction for using the UNL Editor tool to create the semantic graph representation of the sentences. In order to use the tool, the user will have to sign in the UNL web then access the UNL Editor via UNL dev application (The UNL Integrated Development Environment). Reference [19] shows the advantages of the UNL Editor being a web application:

- no installation and updating is required
- easy access through the internet
- data is stored remotely, requiring little or no disk space from the part of the user
- easier to get collaboration possibilities and make contributions

Figure 1 describes the steps for reaching the semantic graphic representation. Within the UNL Editor Frame work, the process of decision making is completely human: the user uploads the text to be analyzed; selects the corresponding IDs; relate nodes through creating semantic relations; and assigns attributes to nodes. The first step will be the text input and text segmentation followed by concepts selection to create the nodes and adding the appropriate attributes to each node then the final step in order to reach the semantic graph will be linking the created nodes by semantic relations [20].

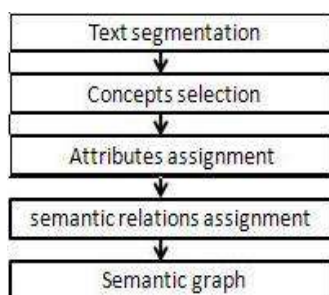


Figure 1: steps for reaching the semantic graph

A. Text Input

After accessing the UNL Editor, the first step is to add the natural language text that needs to be annotated this process could be achieved through two ways; either by selecting the option of “manual text input” in which the user will need to write or paste the source text into an editable area, or by selecting “upload a file” option to upload a file with either text contains UNL, the user wants to modify its content or, to upload plain text contents in order to be converted to the UNL. The contents of the file will be read and parsed into a UNL document format then these documents are presented as projects and are physically stored in the UNL Editor Data Base with the options of removing or downloading these projects, or of adding a new one. Finally, the document will be split into sentences, the UNL adopts some parameters such as “.” for determining the end of the sentences and where the split should be. After the document is split into sentences, the sentences will be ready for the linguistic analysis. After the text has been uploaded and split into sentences, the interface will be divided into two parts; the left pane exhibits the previously saved documents in the upper part while the lower part contains the shared files between the application users, and the right pane contains the sentences that have been segmented. In the case of huge number of sentences which would be saved across many pages, the application provides the user with the ability to navigate between sentences by writing the sentence number in the navigation text box.

Segmented sentences can also be deleted by the "delete sentence button", the user can add a sentence in the document by the "add sentence button". Furthermore, the user could add any comment about the sentences in the comments text box. If there is a problem with the spelling or segmentation, as the application can split “e.g.” since it considers “.” as a delimiter and could segment after it, the user can modify the text by the "editing text button". Then, the sentences are ready to be annotated by using the UNL Editor; the user will have to use the "graph drawing button" to start annotation (see figure 2).

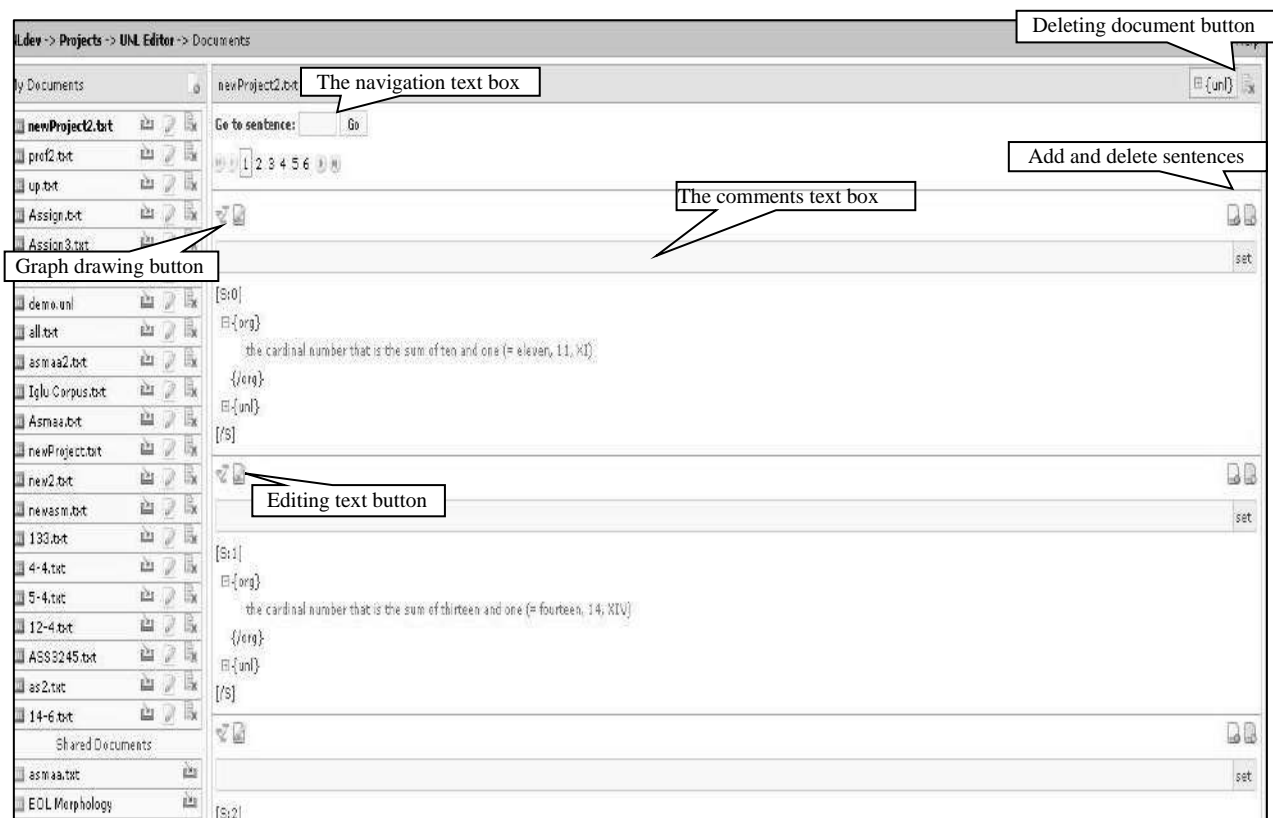


Figure 2: The UNL Editor interface presenting the segmented text

B. Nodes Creation

The first step for annotating a natural language text includes selecting the corresponding ID for each concept, and choosing the appropriate attributes the concepts need in order to complete the meaning of the concepts the sentence contains.

The first step for annotating a natural language text is determining which of the lexical items constituting the sentence represent concept and which do not; usually auxiliary verbs, model verbs and articles are not regarded as concepts and are being

represented by attributes, also the user should determine which constituents represent a compound word and which do not, for example "White House" it could mean the American presidential House or simply a house that is painted white, it is up to the user to decide according to the meaning. Furthermore, compound words may be separated by other units, for instance "look up" is a compound verb that could be separated as in "look the dictionary up". In this case, the user will have to determine the words that represent the intended sense and that should be included in a single node. Only after determining the concepts of the sentence, the user could create the nodes and choose the corresponding IDs.

The option of editing nodes is provided in the interface as after creating a node, the user may discover that this node is not needed in annotation, and needs to be deleted so the option of deleting nodes is provided through a button for deleting nodes "delete node button". Another button is provided for duplicating the nodes "clone node button" as some situation requires duplicating the same node as in the case of ellipsis; the omission from a sentence or other construction of one or more words that would complete or clarify the construction [21]. In a sentence as "I'm leaving and so does he" which means that "I'm leaving and he (is leaving) too" the node "leave" would have to be duplicated in order to represent the entire semantic graph of the sentence, and the attribute "@ellipsis" will have to be assigned to the node. Figure 3 illustrates the buttons needed in the process of creating the nodes.

1) *Concept Selection:* There are three possibilities for looking up the concept when working with the Graph Editor, provided through three tabs that enable the user to choose the method he believes the most appropriate. These three tab ranges from the most general to most specific dictionaries, the first tab is the concepts tab which enables the user to choose senses from the general dictionary uploaded from the WordNet, the second tab is the memory tab to choose from other previous users selections from the WordNet and the final tab is the dictionary tab in which the user uploads his own dictionary:

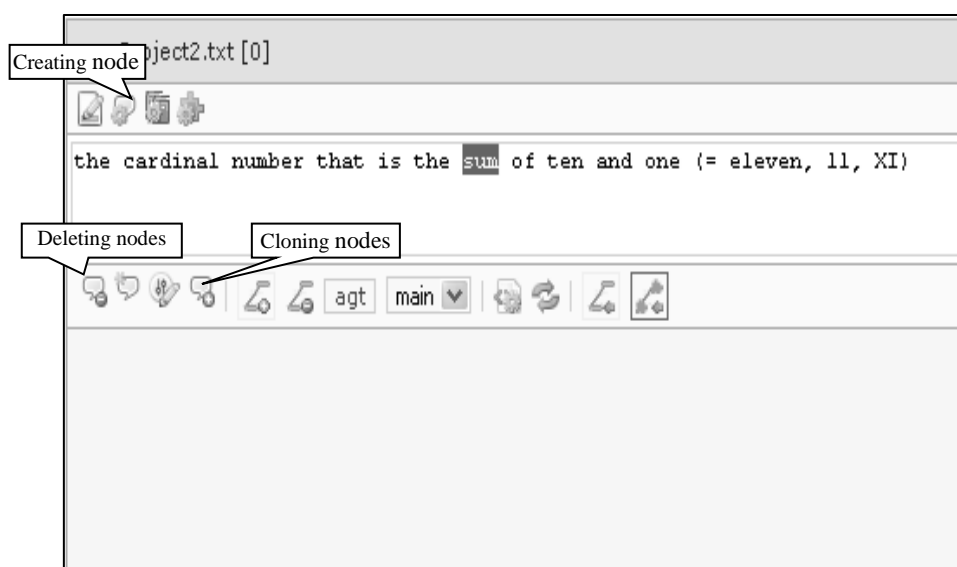


Figure 3: Creating, deleting and cloning the nodes

Concepts tab

This tab matches the lexical items included in the sentence with the concepts extracted from WordNet 3.0. In figure 4, the word "boy" is matched to all the different concepts that could be expressed with the lexical item "boy". In order to obtain a more precise idea about the matching concepts, more details are shown at pointing the mouse on each concept. A light preview appears containing; a distinct ID represented as a nine digit number, an abstract meaning (the gloss), a set of synonyms (the synset), the corresponding part of speech, the frequency and in some senses examples are shown. Moreover, the UNL Editor provides a filtering option in order to facilitate the process of searching; Users are able to search according to the part of speech either it is a noun, proper noun, verb, adjective, participle (A lexical item, derived from a verb, that has some of the characteristics and functions of both verbs and adjectives) or adverb, and for more flexibility there are three search options, the search could be performed according to specific word or string or number.

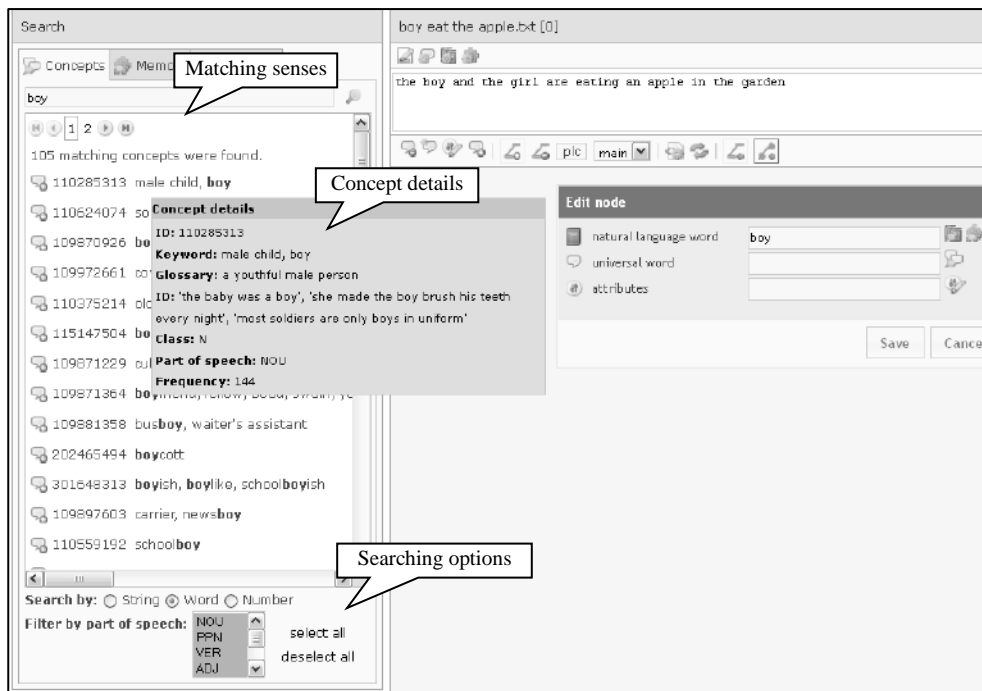


Figure 4: Searching for concept

If the user could not find the corresponding ID for the word, as in the case of names or websites or etc., the tool enables the user to handle this word as a temporary concept by putting that word between double quotes and it would be regarded as a node. The user may face other problem while creating the node as he may not find the appropriate sense for the lexical item; also he can add the node as a temporary concept, but to be added to the dictionary in the future.

Memory tab

This tab displays the dictionary Lookup memory that has the ability to store, retain, and recall nodes accumulated by all users who has used the UNL Editor as a tool to analyze natural language documents, the results show the matching concepts that were found. Unlike the results of the concept tab, the results displayed by the memory tab include the attributes that were assigned to the previously used IDs. The results of the memory tab are of a great use as it gives a clear idea about the frequency of usage of the different senses of the same lexical item, as well as it provides the user with a more feasible results since the concepts are accompanied with the needed attributes. Figure 5 shows the limited list of previously used senses of the concept "boy".

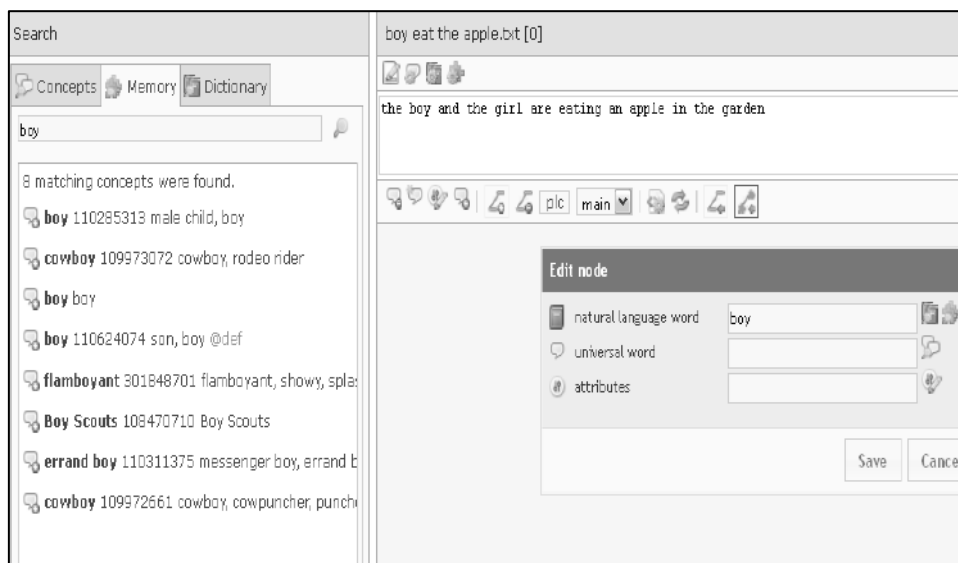


Figure 5: Memory tab

Dictionary tab

This tab offers much flexibility through providing the user with the option of using other dictionaries; the user can use the other dictionaries that exist in the other applications of the UNL web such as the dictionary of the EUGENE application or the dictionary of the IAN application or, he can upload his own dictionary provided that it conforms with the UNL dictionary format. This tab enables the user to create his own dictionary, thus creating the opportunity of having a specialized dictionary for specialized usage. Figure 6 shows the IAN application dictionary.

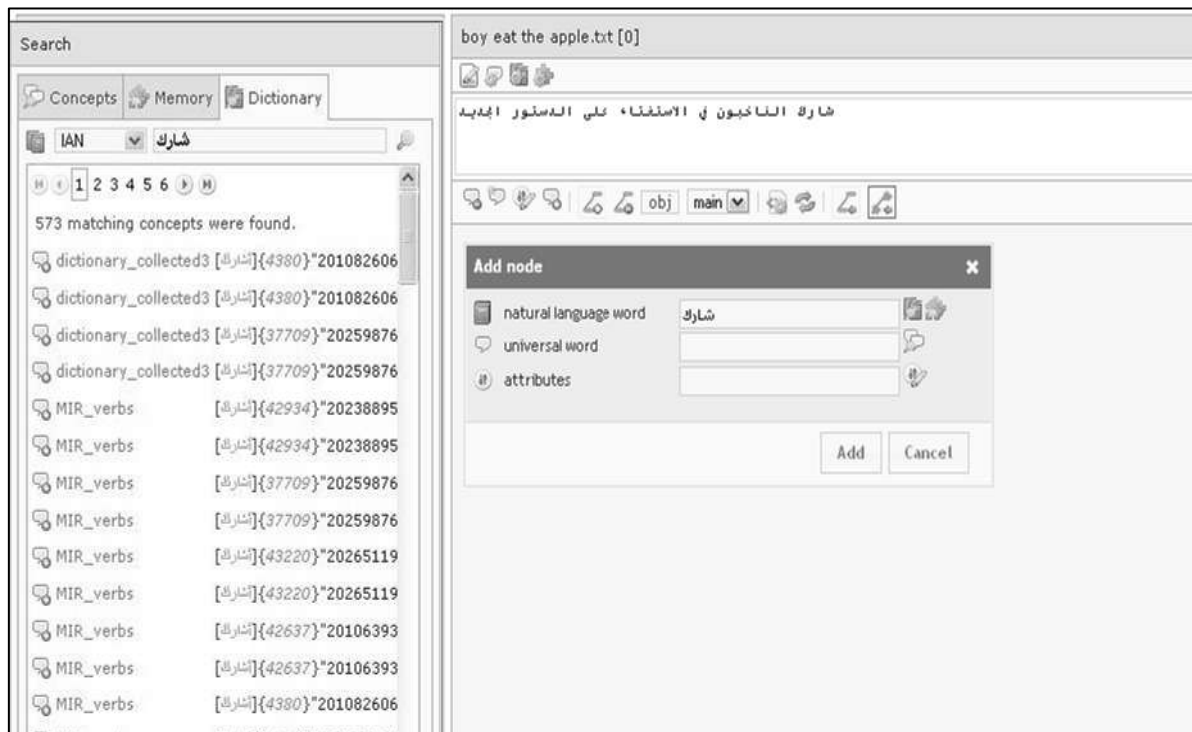


Figure 6: Dictionary tab

2) *Attributes Assignment*: After selecting the appropriate senses for each lexical item, some pieces of information will still be missing and need to be stated for each node in order to represent the whole meaning of the semantic network of the sentence. The UNL Editor provides a comprehensive set of attributes in order to convey these extra pieces of meaning, the added attributes have to be from the fixed list that the application has provided [22]. The process of adding the attributes is manual in the sense that the user of the tool has to add the attributes by writing them or, by coping them from a list of attributes that is available on the web site as in figure 7. The user can edit the added attributes through "modify UW attributes" button, the user can add or modify or delete any attribute. Furthermore, the UNL Editor has provided a special button for determining the entry called "entry assignment" button, since that the UNL specifications require that every sentence has to contain an entry node that represents the most prominent element in the sentence and that would be the starting point of the semantic graph.

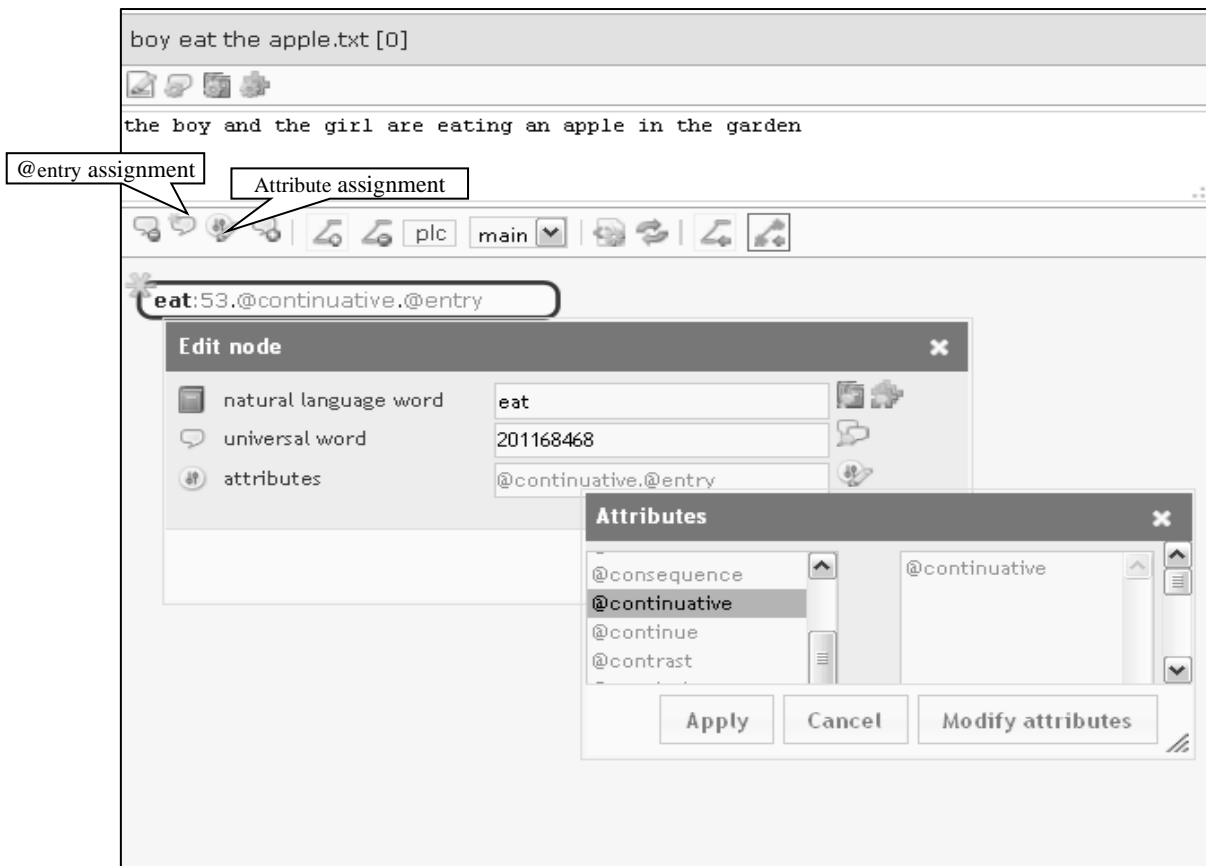


Figure 7: Assigning attributes

C. Linking the Nodes by Semantic Relations

The third and final step of analyzing a natural language sentence using the UNL Editor is the process of creating the semantic relations between the constituent of the sentence. Since that the UNL Editor is especially designed to offer the utmost appropriate environment for providing the analysis of natural language texts, it has provided a toolbar; including different buttons, that are necessary in performing all the needed operations in order to create the required semantic relations between the nodes, all of which is done through a graphic interface. For adding a relation between two nodes, the user could either click on "select relation", a button which consequently opens a list of all the semantic relations provided by the UNL framework from which the user can select the relation he finds most appropriate to convey the intended meaning or, the user could drag one of the two nodes he wants to choose a relation for onto the other where the set of UNL relations will appear and the user will be able to choose the suitable relation according to the meaning. Moreover, in order to modify a relation there has been another button called "remove selected relation" by which the user could remove the relation he selects through clicking on it and then clicking on the button. Every semantic relation used at the UNL framework has a specific direction; meaning that each relation should start from a specific node to go to another node in order to convey the meaning or otherwise the meaning could be distorted [23]. Therefore, a certain button has been provided to swap the direction of the relation after drawing it. It is called "swap selected relations nodes". The user could select the relation he wants to swap its direction by clicking on it then he could swap the relation by clicking on the button. This button has been designed with the intention of saving time and effort. Figure 8 shows a toolbar that includes all the buttons to create the semantic relations between the nodes.

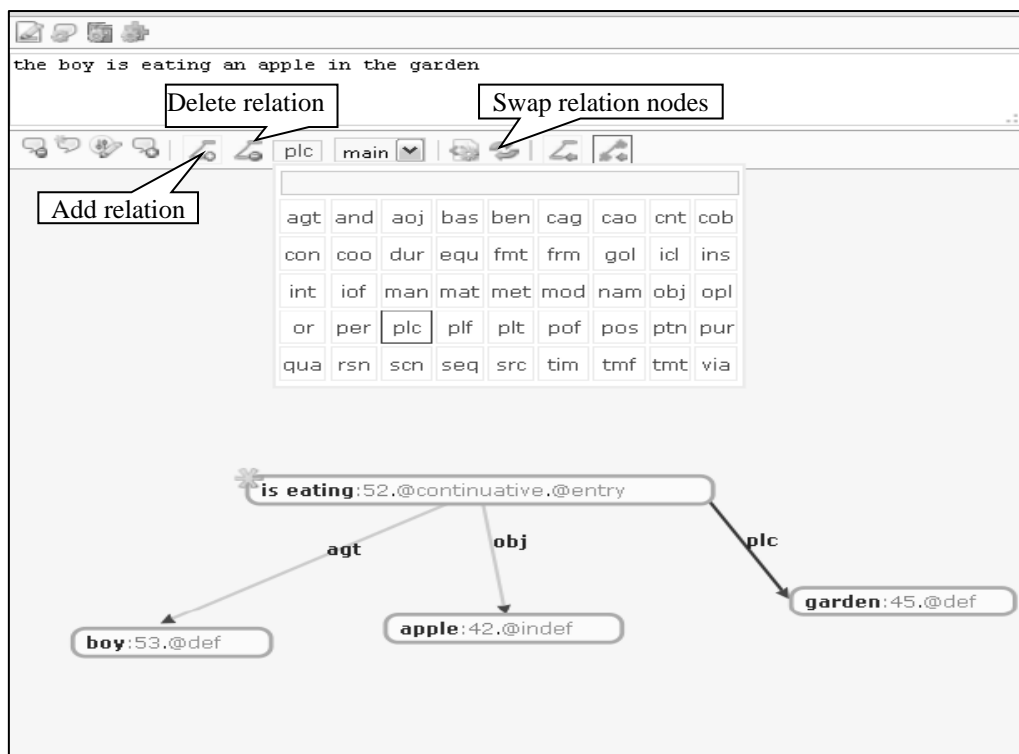


Figure 8: semantic relations between nodes

Creating the scope

The UNL representation is a hyper-graph, which means that it may consist of several interlinked or subordinate sub-graphs. These sub-graphs are represented as hyper-nodes which are named scope which roughly corresponds to the concept of dependent (subordinate) clauses. They are used to define the boundaries between complex semantic entities being represented. Scopes must be used to prevent semantic ambiguities in the following types of clauses:

1- adverbial clauses:

time: her father died (when she was young).

condition: (If they lose weight during an illness), they soon regain it afterwards.

purpose: They had to take some of his land (so that they could extend the churchyard).

reason: I couldn't feel anger against him (because I liked him too much).

consequence: My suitcase had become so damaged on the journey home (that the lid would not stay closed).

concession: I used to read a lot (although I don't get much time for books now).

place: He said he was happy (where he was).

manner: I was never allowed to do things (the way I wanted to do them).

2- adjective clauses:

The vegetables (that people often leave uneaten) are usually the most nutritious.

3- nominal clauses:

subjective: (Why you did that) is a mystery for me.

subjective complement: You can be (whomever you want).

objective: I know (that the weather will be very hot).

Every scope must contain one and only one attribute @entry, to be assigned to the head of the scope. The head of the scope is:

- The main verb, in verbal predicates;
- The subject complement, in nominal predicates;
- The head of the phrase, in phrases and non-finite clauses.

The user can create a scope by selecting the relation that will link the subordinate clause with the rest of the sentence and also selecting "new" from the "clause type" combo box. Then the scope will be created as a new node, as shown in Figure 9. A scope has been created as a new node with the name "01" and has been linked by the selected relation, as in figure 10. All the nodes inside the subordinate clause will be included in the scope, and the scope will be considered as a one unit or hyper node

that clause contains a new entry node since it is regarded as a new sentence embedded in the main sentence. All the embedded nodes will have different color in order to identify them as presenting a single unit as sub-graph.

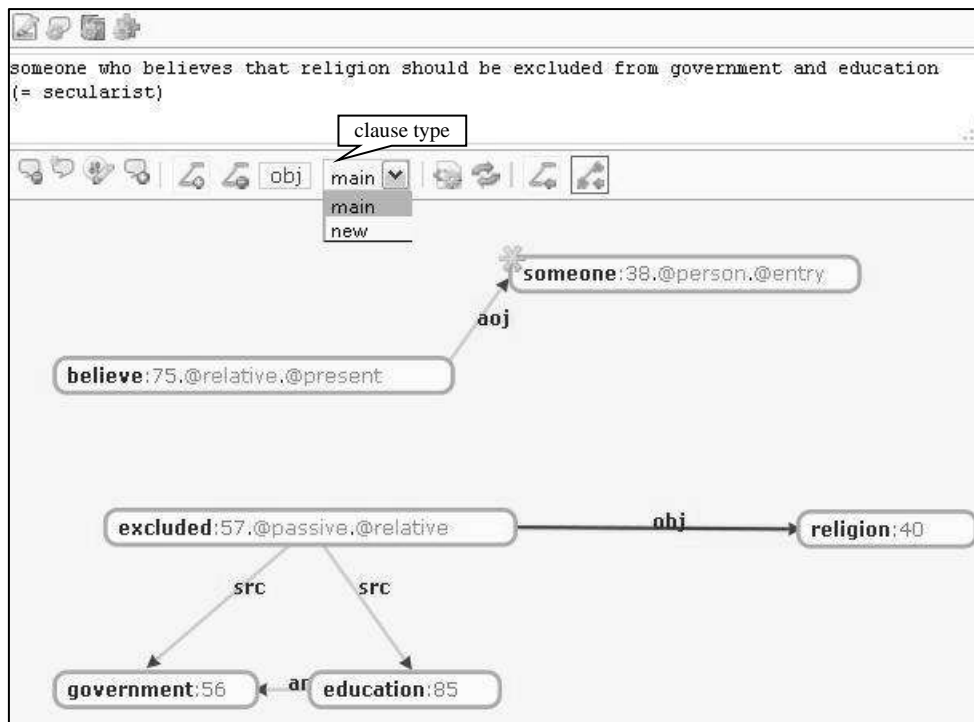


Figure 9: creating the scope

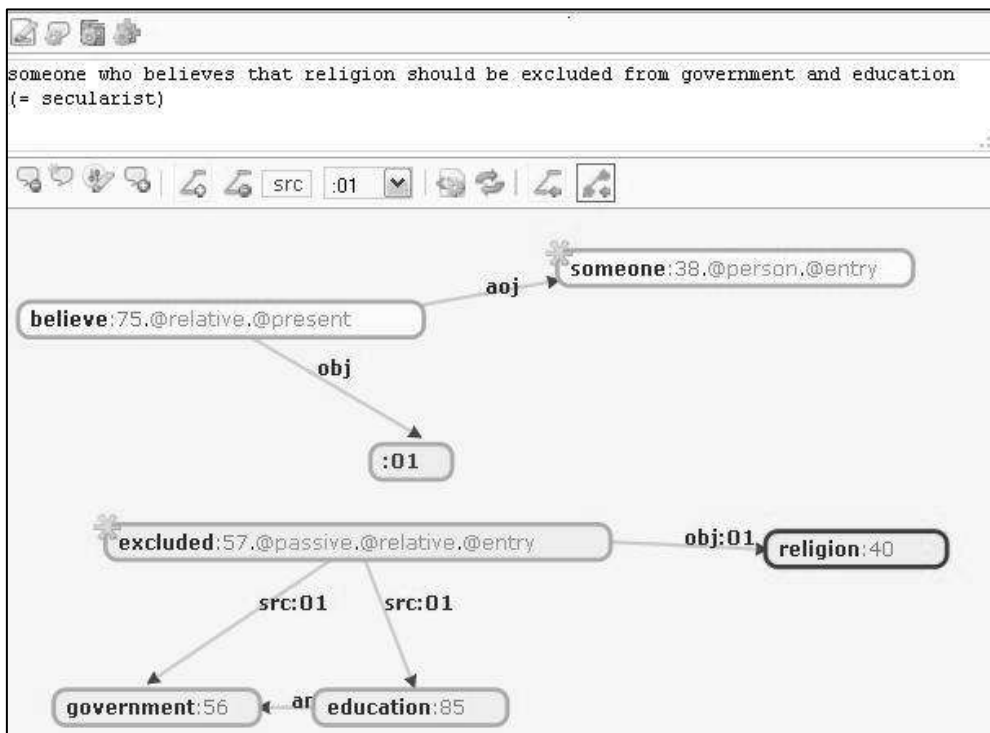


Figure 10: The created scope and the embedded semantic relations

D. Semantic Graph Output

After the relations are created, each sentence can be shown as a graph; the graphs are actually visual editors. They can be modified; nodes are draggable and the relations are clickable as well. The semantic graph could be viewed in two ways; either in

NL view as shown in figure 11, or in Concept view as in figure 12. The output, the semantically annotated text, is downloaded as a text file, making the output a rich material to be used as training data or to be used in other applications. The downloaded file contains the original sentence and the semantic annotated text that is represented as semantic relations between the nodes, each two nodes linked with a relation are inserted between two brackets separated by a comma as shown in figure 13.

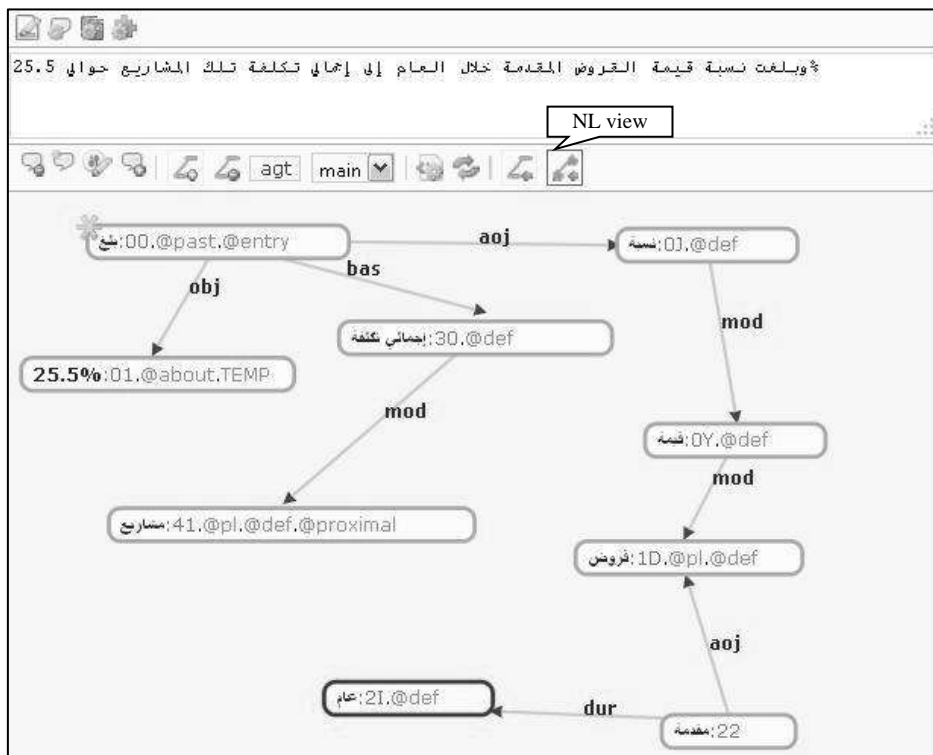


Figure 11: Semantic graph in NL view

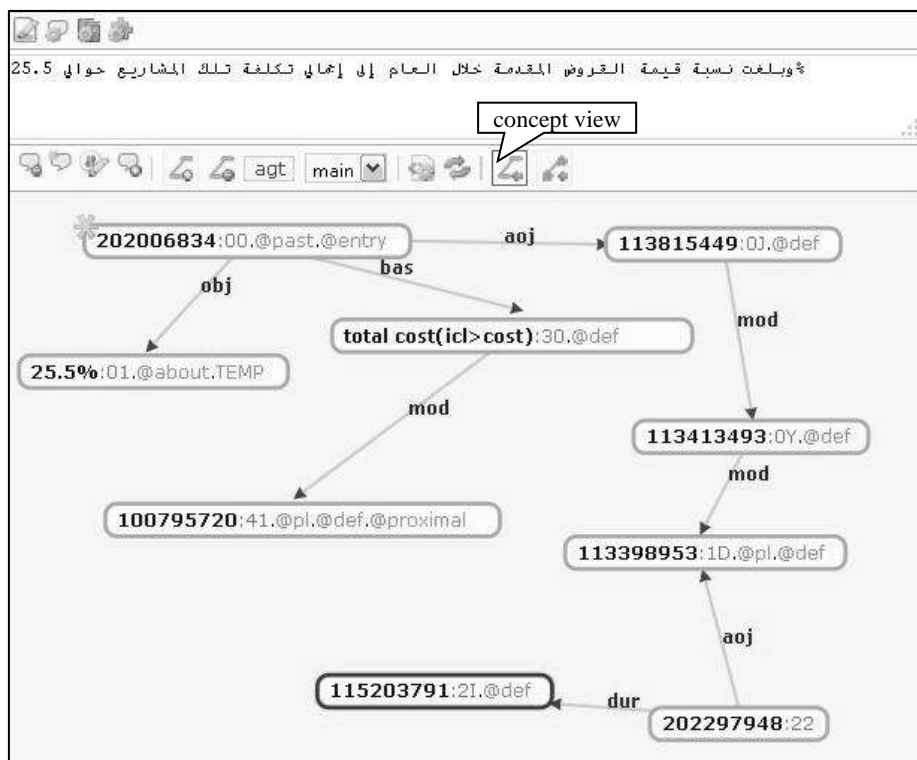


Figure 12: Semantic graph in Concept view

```

[S::1]
{org}
dish baked in pastry-lined pan often with a pastry top (= pie)
{/org}
{unl}
obj(200319886:41.@passive.@relative,107557434:78.@entry)
plc(200319886:41.@passive.@relative,103880531:54)
man(200319886:41.@passive.@relative,400059547:53)
aoj(300258797:39,103880531:54)
mod(103880531:54,107622708:14)
mod(108663860:61.@indef.@with,107622708:58)
man(200319886:41.@passive.@relative,108663860:61.@indef.@with)
{/unl}
{/s}
[S::2]
{org}
drops of fresh water that fall as precipitation from clouds (= rain, rainwater)
{/org}
{unl}
mod(113771404:68.@pl.@spec.@entry,115009326:27)
man(201972298:33.@relative,111494638:52.@as)
src(201972298:33.@relative,109247410:74.@pl)
obj(201972298:33.@relative,113771404:68.@pl.@spec.@entry)
{/unl}
{/s}
[S::3]
{org}
education that results in understanding and the spread of knowledge (= enlightenment)
{/org}
{unl}
aoj(200340704:16.@relative.@present,100883297:58.@entry)
mod(107445896:43.@def.@spec.@in,100023271:63)
and(107445896:43.@def.@spec.@in,105805475:72.@in)
obj(200340704:16.@relative.@present,105805475:72.@in)
obj(200340704:16.@relative.@present,107445896:43.@def.@spec.@in)

```

Figure 13: the Generated UNL in a text file

4 UNL Editor Usage

The UNL Editor performs morphological, syntactic and semantic analysis synchronously. It is able to represent, describe, summarize, refine, store and disseminate information in a natural-language-independent format. It enables people to make their own UNL documents and providing the output in a text file, thus providing analyzed corpus to be used in other applications [24], so we can summarize the usage of the UNL Editor as follows:

- Generating semantic networks to interpret and understand the underlying semantics of the documents.
- Building analyzed corpus which is morphologically, syntactically and semantically analyzed.
- Important for applications such as information extraction, question answering, machine translation, summarization, complex filter and search operations.

5 CONCLUSIONS

This paper presented the UNL Editor as a pioneering effort for providing a tool for semantically annotating natural language texts, which all is done through a graphic interface that allows users to manipulate high-level graphs. After presenting the state of complete lack of such tools, and establishing the urgent need for it due to its importance and the range of applications a semantic annotation tool serves as a basis for. It represented the approach adopted in building this tool and the linguistic theories integrated in designing it pointing out how this approach offered great opportunity to overcome linguistic difficulties, it explained how this tool could be used and stated how feasible its output is. It also, presented the enormous opportunities that UNL Editor offers as a tool for performing data analysis.

REFERENCES

- [1] H. Bunt and Ch. Overbeeke, "A note on the definition of semantic annotation Languages", *Proceedings of the 8th International Conference on Computational Semantics*, pages 268–271, Tilburg, January 2009. c 2009 International Conference on Computational Semantics 2009.
- [2] P. Hitzler, M. Krötzsch, S. Rudolph, *Foundations of Semantic Web Technologies*, Chapman & Hall, 2009.
- [3] A. Giuglea and A. Moschitti, "Semantic Role Labeling via FrameNet, VerbNet and PropBank" *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL ACL 06*, Pages: 929-936, 2006.
- [4] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: A framework and graphical development environment for robust NLP tools and applications", *40th Anniversary Meeting of the Association for computational Linguistics-ACL'02*, 2002.
- [5] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ogniano, and M. Goranov, "Kim - semantic annotation platform", *2nd International Semantic Web Conference-ISWC2003*, pages 834-849, Florida (USA), 2003.
- [6] F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks, "User-system cooperation in document annotation based on information extraction", *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 02)*, Sigenza (Spain), October 2002.

- [7] H. Cunningham, D. Maynard, and Tablan, "Jape: A java annotation patterns engine", Technical report CS--00--10, University of Sheffield, Department of Computer Science, 2000.
- [8] F. Ciravegna, "Designing adaptive information extraction for the semantic web in amilcare, Annotation for the Semantic Web", *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 02 (2002)*, 2003. [9] J. D. Choi, C. Bonial, M. Palmer, "Cornerstone: Propbank Frameset Editor Guideline", Technical Report 01-09, September 28, 2009.
- [10] Fillmore, C. J., and Baker, "Frame semantics for text understanding", *In Proceedings of WordNet and Other Lexical Resources Workshop Held in conjunction with the NAACL Annual Meeting*, 1968.
- [11] Ch. Johnson and Ch. J. Fillmore: "The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure". *In the Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, Seattle WA, pp. 56-62, April 29-May 4, 2000.
- [12] D.R Dowty, *Thematic Proto-Roles and Argument Selection*, Linguistic Society of America, 1991.
- [13] C. Fellbaum, *WORDNET: An Electronic Lexical Database*, The MIT Press, 1998.
- [14] N. Chomsky, *Studies on Semantics in Generative Grammar*, Mouton publisher, 1972.
- [15] J. Pustejovsky, *The Generative Lexicon*, MIT Press, 1995.
- [16] G.A. Miller, "WordNet: A Lexical Database for English", *Proceedings of the 11th ACM conference on Computer and communications security*, Vol. 38,1995.
- [17] A. Wierzbicka, *The Semantics of Grammar*, John Benjamins Publishing Company, 1988.
- [18] D. Pisoni and R. Remez, *The Handbook of Speech Perception*, Blackwell Publishing Inc., 2004.
- [19] Web Apps vs Desktop Apps site: <http://valums.com/web-apps/>
- [20] M.Zhu and H.Uchida, "UNL annotation", *UNL Center, UNDL Foundation specifications and manuals*, 2003.
- [21] Sh. Lappin, *The interpretation of ellipsis. In The handbook of contemporary semantic theory*, Blackwell Publishing Inc, 1996.
- [22] UNL attributes available from: http://www.unlweb.net/unlarium/dictionary/export_attributes.php.
- [23] UNL Relations available from: <http://www.unl.org/unlsys/unl/unl2005/relation.htm>.
- [24] V. Uren, Ph. Cimiano, J. Iria, S. Handschuh, M.Vargas-Vera, E. Motta, F. Ciravegna, "Semantic annotation for knowledge management: Requirements and a survey of the state of the art", *Journal of Semantic Web*, 2005.

اللسانيات الحاسوبية من منظور مجتمع المعرفة

د. نبيل على

خبير اللسانيات الحاسوبية

ملخص البحث:

تمر اللسانيات الحاسوبية بنقلة نوعية حاسمة حيث تسعى إلى الارتقاء من معالجة الظواهر اللغوية السطحية على مستوى الحرف وبنية الكلمات وتراكيب الجمل إلى معالجة الجوانب الدلالية الكافية في النصوص من معان ومفاهيم وعلاقات سياقية ومنطقية.

تتطلب هذه النقلة النوعية إعادة النظر في المعالجات اللغوية الصرفية والمعجمية والنحوية وتأسيس العلاقة بين البنى النحوية والصيغ المنطقية المناظرة لها.

على الجانب المعجمي تتطلب النقلة النوعية المذكورة بناء قواعد بيانات معجمية على أساس المفاهيم لا المفردات كما في المعاجم التقليدية وكذلك بناء الأنطولوجيات القائمة على هرميات المفاهيم من أجل خدمة الويب الدلالي.

تتناول الورقة كذلك مسارات التواصل بين مجتمع المعرفة وحوسبة اللغة العربية عبر عدة مستويات تغطي الجوانب الثقافية واللغوية والتربوية والإبداعية.

A Comparative Corpus-Based Study of the Complement Structure of the Verb “Said” and “Qala” in English and Arabic

Ateka Nasher^{*1}, Sameh Al-Ansary^{**2}, and Shadia El-Soussi^{***3}

**English Language and Literature Department, Linguistics Branch, Faculty of Arts, Alexandria University, Alexandria, Egypt.*

¹nasher_ateka777@hotmail.com

***Phonetics and Linguistics Department, Faculty of Arts, Alexandria University, Alexandria, Egypt.*

²sameh.alansary@bibalex.org

****Institute of Applied Linguistics, Faculty of Arts, Alexandria University, Alexandria, Egypt.*

³selsooussi@hotmail.com

Abstract — The aim of this paper is to present the differences and the similarities between the verb complements of the past tense of the verb “to say” (said) in English and the verb “qala” in Modern Standard Arabic (MSA). The linguistic approach used in this study is the conjunction of the Immediate Constituents(IC) and the Functions and Categories Alternation. This approach has been obtained from the British Component of the International Corpus of English (ICE-GB), and is applied to the MSA sample analysis in order to validate this contrasting study.

According to the frequency of the different types of the verb complement of both verbs, it is obvious that the usage of the clause (CL) introduced by the noun phrase (NP) in direct speech is more frequent than the other usage of the verb complement after “said”(39.68% of the total occurrence of said). In contrast, the most frequent occurrence of “qala” is with Nominal Sentence (NS) introduced by Inna. It occurs 86.1% of the total occurrence of “qala”. Thus, the high frequency shows the greater usage of Inna after “qala” whereas low frequency of other types (14%) point rather to marginal usage.

1 INTRODUCTION

Corpora may play a significant role in the study of language. There are two major strengths of the corpus-based technique to linguistic analysis. Firstly, text corpora provide huge databases of naturally-occurring discourse, enabling empirical analyses of the actual patterns of use in a language. Secondly, when this empirical data is combined with (semi-)automatic computational tools, the corpus-based approach enables analyses of a scope not otherwise achievable [1]. Corpora have been introduced into many linguistic disciplines; and have succeeded in opening up new areas of research or bringing new insights to traditional research questions. For instance, numerous studies describing the formal variants and functions of particular grammatical constructions have been based on analysis of large text corpora (see the bibliography compiled by Altenberg [2], containing approximately 650 references to studies based on corpora). Recent book-length treatments of this kind include Tottie's [3] analysis of negation in English, Mair's [4] analysis of infinitival complement clauses, and Meyer's [5] study of apposition.

Regarding Arabic language, some researchers have used corpus-based approaches in their studies; for example, Al-Motwakil [6], explored the descriptive capabilities of functional grammar (FG) with respect to syntax and the features of Arabic language. Fassi Fehri [7] adopted an approach by which described the sentence structure in Arabic. He emphasized the rule of the lexicon to make transformations more realistic or at least to restrict the number of transformation. Ditters [8] presented a formal approach to Arabic syntax: the noun phrase and verb phrase. In (2000), he [9] presented a corpus-based study in basic structures of Modern Standard Arabic syntax in terms of function and categories. Al-Ansary [10] presented a powerful strategy in which he used the IC with function and categories alternation approach for comparing the NP structure of spoken and written Modern Standard Arabic.

2 METHODOLOGY OF THE RESEARCH

The choice of methods stems from the nature and the objectives of the research. A contrastive corpus-based approach is followed to identify the similarities and the differences of the syntactic structure between the verb complement of the verb *said* in English and *qala* in MSA. Corpus analysis can be broadly categorized as consisting of both qualitative and quantitative analyses. Since corpus analysis encompasses both qualitative and quantitative analyses, the former is implemented through verb complement analysis of *said* and *qala* and the latter is applied through statistical analysis. More specifically, for the purpose of qualitative analysis the syntactic formalism that is adopted is the Immediate Constituents (IC) with Functions and Categories alternation.

3 CORPUS DESCRIPTION

Concerning English data, the written part of the British component of the International Corpus of English (ICE-GB) is obtained. For Arabic data, Al-Ahram 99 from the online corpora, the Arabic corpus (arabiCorpus) is used. This research focuses on the verb *to say* (*said*) in English and the verb *qala* in MSA, as they are both used in reported speech.

The ICE-GB has been fully tagged and parsed, and is being released simultaneously with The ICE-GB Corpus Utility Program (ICECUP), a text-analysis program that fully exploits the extensive grammatical annotation that the ICE-GB contains. Taken together, ICECUP and ICE-GB provide the corpus linguistic community with a powerful resource for the analysis of present-day British English [11], [12].

Regarding MSA data, an online Arabic corpus 'arabiCorpus' was used. Unlike the ICE-GB, the Arabic corpus is a free online corpus but is an untagged meaning that the syntactic analysis presented by the tree diagram was done manually. Nevertheless, it is designed to facilitate research. It is an untagged corpus, but the part of speech can be chosen because its program can perform a morphological analysis. What the program does is find every item in the corpus that matches the search string you type in, and then it filters those results based on the part of speech you choose. The program does not do any analysis of the surrounding context, only of the form itself [13].

4 THE UTILITY OF USING THE IC WITH FUNCTIONS AND CATEGORIES ALTERNATION IN ANALYSING THE VERB COMPLEMENT OF THE VERBS SAID AND QALA

The IC approach is based on the constituency relations between the different elements that comprise the sentence. The constituents are divided into parts until reaching the smallest indivisible unit, the morpheme [14]. IC analysis does not take into account the functions of any given constituents or class of constituents– or indeed the sentences as a whole. It therefore needs to be integrated with functions and categories alternation to reveal the relationships between the components of the sentence; by labelling grammatical functions, we can show what part each component is playing in the overall structure.

Thus, the linguistic approach using IC with functions and categories alternation is the most suitable linguistic approach for this research because it reveals the relationships between the verbs *said* and *qala* and their complements as well as between each element inside the verb complement. The following examples illustrate the conjunction of these approaches in analyzing the verb complement (VC) of both verbs *said* and *qala*:

1. “Anyone can make a mistake,” Brett said desperately (ICE-GB: W2f-001<90:1)

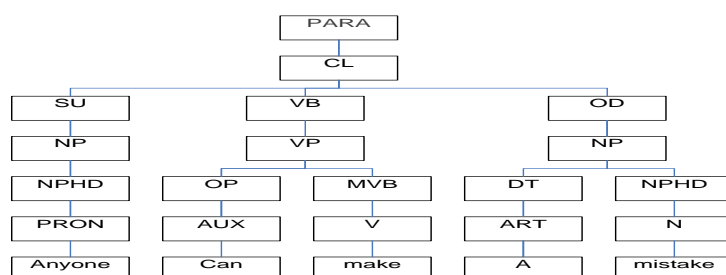


Figure 1: The Syntactic Structure of the VC of the Verb *Said* in Example 1

In this example, the PARA function indicates direct speech in the ICE-GB. At the higher level the PARA FUNCTION (the verb complement) is realized by the CL category. At the lower level, direct speech is composed of the functions: SU, VB and OD, which are realized by the categories: NP, VP and NP.

At the next lower level, each of these phrases is further broken down into further elements. For example, the NP is reduced into the function NP head (NPHD), and likewise for the other phrases. As this example shows the alternation of functions and categories continues until each lexical item is accounted for. This approach can be applied to MSA, for example:

2. قال إن الحكمة تقتضي أن تكون الكلمات في وقت الأزمات محسوبة
 /qa:la ?inna l?ikmata taqtad⇒i: ?an taku:na lkalima:tu fi: waqti
 l?azama:ti ma?subah/
 He said: "It is wise to watch your words in times of crisis."

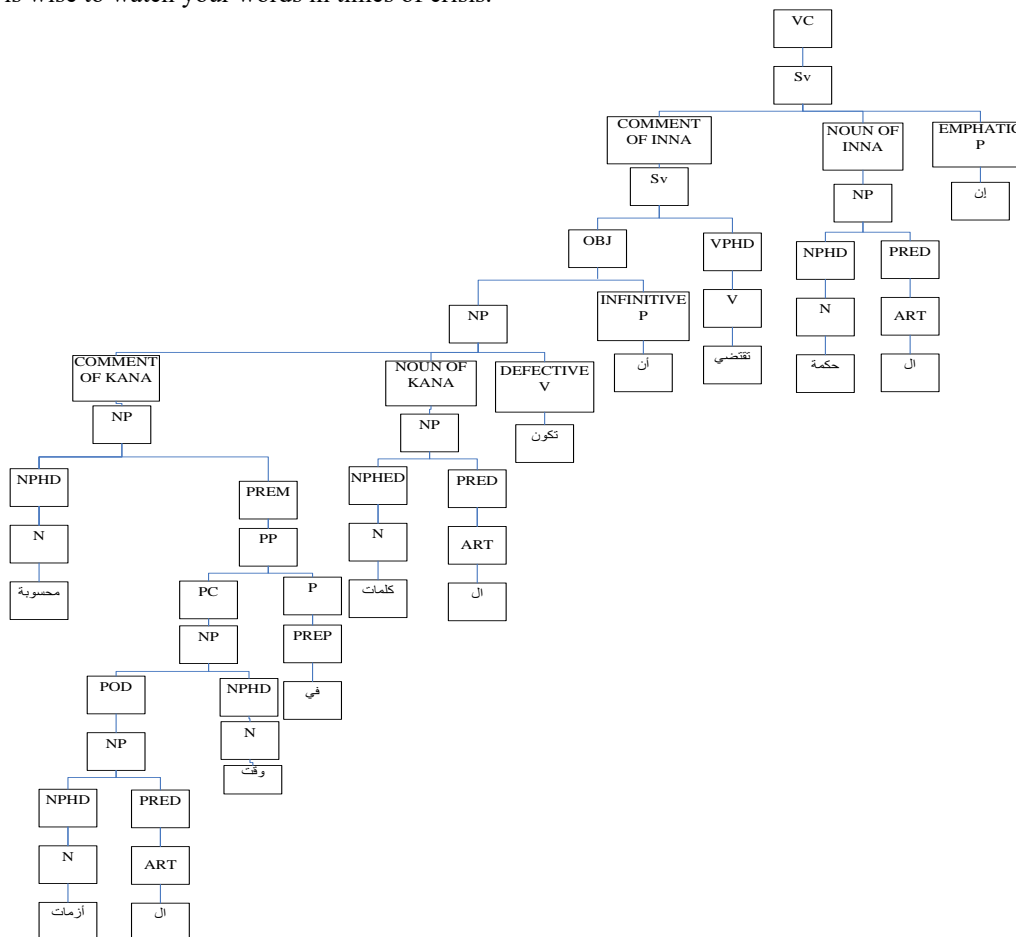


Figure 2: The Syntactic Structure of the VC of the Verb Qala in Example 2

At the higher level, the VC is realized by the NS that is composed of *Inna*, its Noun and its Comment. At the lower level, each of these phrases is further broken down. For example, the VP is broken into the VP head (VPHD) تقتضي and the object(OBJ), which is realized by the CL "إن تكون الكلمات في وقت الأزمات محسوبة". The whole sentence, which is composed of *Inna*, its noun, and its comment functions as a direct object of the verb *qala*. For more information about the syntactic structure of the other VC in English and MSA the reader is referred to [15]. Thus far, the syntactic structure of the verb complement was analyzed. The other objective of this study is to determine the frequencies of different verb complements of *said* and *qala* in English and MSA.

5 TYPES OF VERB COMPLEMENTS OF THE VERB SAID

The verb *said* was found 565 of the time in the ICE-GB. Table I gives an idea of the proportions of the verb complement of the verb *said* found in the corpus. It surveys the whole corpus.

TABLE I
TYPES OF VERB COMPLEMENTS OF THE VERB SAID

The Syntactic Behaviour of the Verb <i>said</i>	Number of Tokens	Number of Tokens Per Million	Percentage %
1. <i>said</i> + Direct Speech (PARACL)	234	585	47.36
2. <i>said</i> + Indirect Speech (OD)	153	400	32.30
3. <i>said</i> + Direct Speech (PARA NONCL)	52	130	10.53
4. As + SUBJ + <i>said</i>	16	40	3.24
5. <i>said</i> + NP (anaphoric reference)	7	17.5	1.42
6. <i>said</i> as Parenthesis CL	32	80	6.48
Total	494	1235	100

Pertaining to the verb *said*, the least frequently occurring types of complements include phrases such as Formulaic Expressions “he said “hi””, CL introduced by the infinitive CL, and others, which make up about 20% of the complements. The most frequently used verb complement is direct speech, accounting for nearly half of all examples. The next most frequent type is the verb complement used is indirect speech, in which the conjunction *that* is deleted, accounting for 18% of all occurrences. The third most frequently occurring complement is the clause introduced by the conjunction *that*, occurring 13% of the time. These numbers may come as surprise to non-native speakers because they illustrate that the conjunction *that* is more often omitted in indirect speech than it is included. This is particularly true for Arabic speakers who are accustomed to seeing *Inna* follow the verb *qala* as we will observe in the next section.

6 TYPES OF VERB COMPLEMENTS OF THE VERB QALA

Because of its high frequency, *qala* is considered the most widely used verb of all reporting verbs. It occurs 38,188 of the time in the Arabic corpus, but this number is reduced to only 25,914 in the examined data after excluding Colloquial and Classical Arabic examples in order to focus on MSA. Table II shows the different types of verb complements of the verb *qala* with the frequency of occurrence of each type.

Regarding the verb *qala*, the least frequently occurring types of complements, include sentences such as: NS composed of topic and comment, VS, and stylistic sentence as: conditional sentence Interrogative sentence, etc, occur only 14% of the time. The low frequency of these complements points to their rather marginal usage. Whereas the most frequently occurring complement of *qala* is the NS introduced by *Inna* which occurs 86.1% of the time. The high frequency of *Inna* demonstrates its prominence as a complement of *qala*.

The high frequency of *Inna* is due to the fact that it is used in both direct and indirect speech. Since *Inna* may have formed part of the original utterance, it is never absolutely certain whether it is part of the original sentence in direct speech or acting as a conjunction, like the English *that*, in indirect speech. Thus, It is impossible to determine the ratio of direct to indirect speech in sentences introduced by *Inna* in MSA.

TABLE II
TYPES OF THE COMPLEMENTS OF THE VERB QALA

The Syntactic Behaviour of the Verb <i>qala</i>	Number of Token	Number of Token Per 1 Million	Percentage %
1. <i>qala</i> + NS introduced by <i>Inna</i>	22,309	1354	86.1
2. <i>qala</i> + NS (topic + comment)	783	47.52	3.02
3. <i>qala</i> + VS	594.6	36	2.29
4. <i>qala</i> + Imperative S	396.4	24	1.53
5. <i>qala</i> + Defective Verbs	243	14.75	0.94
6. <i>qala</i> + Reply P	241	14.63	0.93
7. <i>qala</i> + Conditional S	216	13.11	0.83
8. <i>qala</i> + NS Introduced by PP+ <i>Anna</i>	202	12.26	0.78
9. <i>qala</i> + Interrogative S	190	11.53	0.73
10. <i>qala</i> + Anaphoric or Cataphoric reference	103	6.25	0.40
11. <i>qala</i> + <i>Laa</i> that Denies the Whole Genus	63	3.82	0.24
12. <i>qala</i> + Vocative S	39	2.37	0.15
13. <i>qala</i> + Oath S	1	0.06	0.004
14. <i>Kama</i> + <i>qala</i>	533	32.35	2.06
Total	25,914	1,572.83	100

7 TYPES OF VERB COMPLEMENTS OCCURRING IN BOTH ENGLISH AND MAS

Some verb complements correspond to one another in English and MSA, and therefore occur in both corpora, but with different frequencies. These include direct speech introduced by reply particle, the interrogative CL, direct speech introduced by the VP in its imperative form, anaphoric reference, and the conditional CL.

The verb complements introduced by the VP only occur in the imperative form in English, but they occur in both indicative and imperative forms in MSA. Because only the imperative form is common to both languages, the comparison only discussed the frequency of imperative verbs, ignoring Arabic indicative forms, which lack an English counterpart. As shown in figure 3, these verbs comprise 2.83% of the complements in the ICE-GB, while in Al-Ahram, they comprise 1.53%. Other marginal verb complements include: the CL that is introduced by an interrogative pronoun in direct speech, occurring 3.24% of the time in the ICE-GB and 0.73% in Al-Ahram; the CL that is introduced by the conditional particle occurring 0.2% of the time in the ICE-GB and 0.83% in Al-Ahram; the NP that is used as an anaphoric reference such as the demonstrative pronoun *that* in English and *ذلك* in MSA, occurring 1.42% of the time in the ICE-GB and 0.40% in Al-Ahram.

The NONCL that contains REACT function as *yes/no* words in the ICE-GB appears in Al-Ahram as a verb complement introduced by the reply particle. It makes up 3.64% of the ICE-GB and 0.93% of Al-Ahram. The final pattern that occurs in both corpora is one in which the verbs *said* and *qala* are combined with the preposition *as* and its Arabic counterpart *kama*. It occurs 3.24% of the time in the ICE-GB and 2.06% in Al-Ahram.

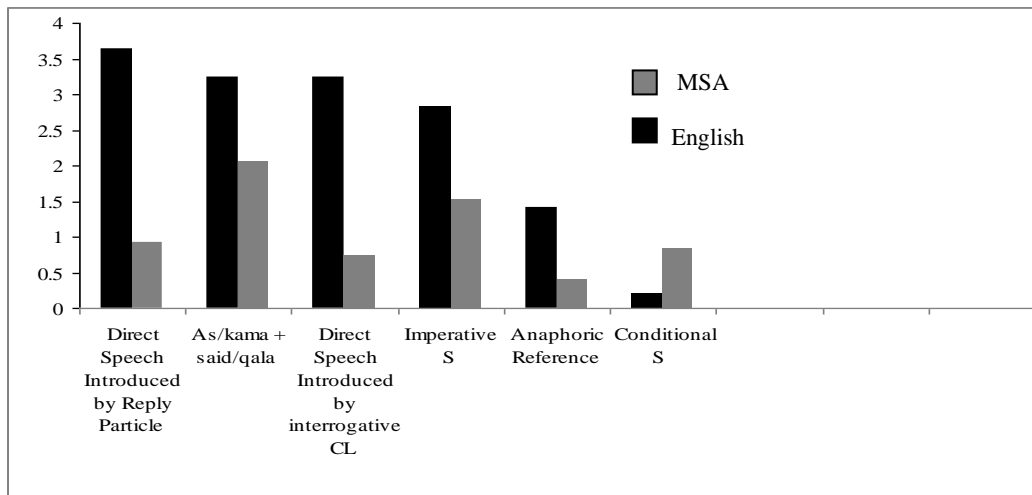


Figure 3: Types of Verb Complements Occurring in both English and MAS

8 TYPES OF VERB COMPLEMENTS OCCURRING IN ONE LANGUAGE AND NOT THE OTHER

Some verb complements are exclusively used in one corpus and not the other. As shown in figure 4, in the ICE-GB, one such type includes verb complements that are part of the NONCL in direct speech, excluding the aforementioned REACT function; it comprises 10.53% of the total. Another verb complement exclusive to the ICE-GB is the CL introduced by the particle *to*, (i.e. the infinitive form of the verb), occurring 0.41% of the time. The final pattern that is particular to the ICE-GB is that in which *said* occurs in a parenthetical CL; this occurs 6.48% of the time.

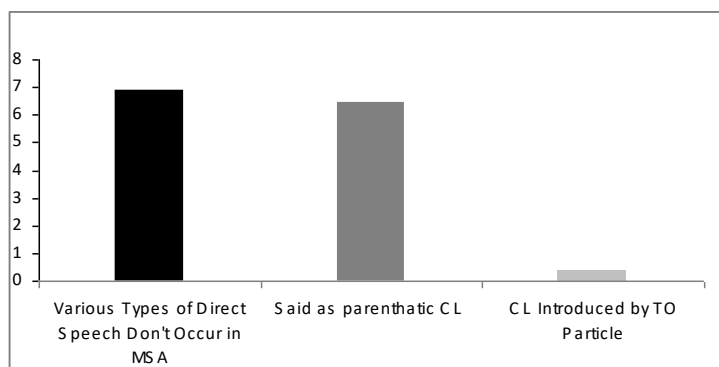


Figure 4: Types of Verb Complements Occurring in English Only

Types exclusive to Al-Ahram are sentences introduced by the vocative particle, occurring 0.15% of the time, as well as the oath sentence, which only occurs 0.2% of the time. These types are presented in figure 5.

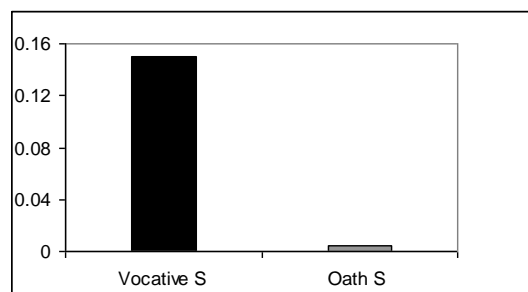


Figure 3: Types of Verb Complements Occurring in MSA Only

Due to the different syntax of Arabic, the structure of the NS differs completely from its English counterpart. Therefore, it is not possible to find English equivalents to other particles that follow the verb *qala* in MSA (e.g. the defective verbs, the *Laa* that denies the whole genus, etc.). However, all other types of verb complements after *qala* can be collectively considered a marked case, since they only comprise 14% of the whole corpus in contrast to the 86% of NS's introduced by *Inna*.

9 CONCLUSION

The data obtained through this research provide more explicative and descriptive insights into the nature and structure of verbal complements and how they correspond between two unrelated languages. After quantifying the frequency of the different structures found in the two languages, commonalities that were previously buried under superficially different syntactic structures become evident. Thus these two languages' disparate methods of expressing similar ideas reveal their deep similarities in ways that intuition alone might fail to discover.

Information obtained in this study is of benefit to translator when translating from English into MSA and vice versa. For example, when translating from English into MSA, translators must be aware that the usage of *Inna* is not always the same as the conjunction *that* in English. That is, when it occurs as a part of the original speech it functions as an emphatic particle, but when *Inna* introduces indirect speech it is an equivalent to the conjunction *that* introducing indirect speech in English. Thus, a sentence introduced by *Inna* can be translated either as direct speech or as indirect speech. Conversely, a translator working from Arabic to English should use the variety of available clauses instead of always copying the structure of the Arabic and using indirect speech introduced by *that*. This research demonstrates that, although it is grammatically equivalent to the most common structure found in Arabic, indirect speech introduced by *that* is comparatively rare in formal written English. Therefore, translators have to bear in mind of the different structures of direct and indirect speech in both the source and target languages. Finally, computational linguists working with formal grammar may benefit from the results produced through this study, i.e. the syntactic structure of the verb complement of the verb *said* in English and *qala* in MSA, as they may enhance the quality of an English and MSA parser. In addition, the linguistic description of the sentence in English and MSA can be represented formally through building a Natural Language Processing tool (NLP).

REFERENCES

- [1] C. Meyer, *English Corpus Linguistics: An Introduction*. Cambridge: Cambridge University Press, 2004.
- [2] B. Altenberg, A Bibliography of Publications Relating to English Computer Corpora in S. Johansson and A-B. Stenstrom (Eds.) 1991: *English computer Corpora: Selected Papers and Research Guide*. New York: Mouton, 1991.
- [3] G. Tottie, *Negation in English Speech and Writing: A Study in Variation*. San Diego: Academic Press, 1991.
- [4] C. Mair, *Infinitival Complement Clauses in English: A Study of Syntax in Discourse*. Cambridge: Cambridge University Press, 1990.
- [5] C. F. Meyer, *Apposition in Contemporary English*. Cambridge: Cambridge University Press, 1992.
- [6] A. Al-Motawakil, Topics in Arabic: Towards a Functional Analysis. In Bolkestein, A.C. de Groot and Mackenzie(Eds.), *Syntax and Pragmatics in Functional Grammar*. Dordrecht : Foris Publication, 1985.
- [7] A. Fassi Fehri, *ʔallisanijja:t wa alluΓa ʔalʔarabijja*. Beirut, 1986.
- [8] W. E. Ditters, *A formal approach to Arabic Syntax: the noun phrase and the verb phrase*, Amsterdam:Nijmegen University, 1992.
- [9] W. E. Ditters, "Basic Structures of Modern Standard Arabic in Terms of Function and categories," in *Proc. International Conference on Artificial and Computational Intelligence for Decision Control and Automation in Engineering and Industrial Applications. Natural Language Processing Panel*, pp. 83,22-42, Tunisia: Monastir, March, 2000.
- [10] S. Al-Ansary, "A Comparative Corpus-Based Study of Spoken and Written Modern Standard Arabic", Doctoral dissertation, Alexandria University, 2002.
- [11] G. Nelson, *The International Corpus of English: The British Component ICE-GB*. London: University College London, 1998 .
- [12] G. Nelson, (1990), *The International Corpus of English*, Available from: <http://www.ucl.ac.uk/english-usage/ice/>.(accessed 20 December 2004).
- [13] D. Parkinson, (2006). arabiCorpus, Available from: <http://arabicorpus.byu.edu/search.php> , (accessed 6 March 2006).
- [14] F.R. Palmer, *Grammar*. London: Penguin, 1990 .
- [15] A. Nasher, "A Contrastive Corpus-Based Study of the Syntactic Behaviour of the Verbs "said" in English and "qala" in Modern Standard Arabic," An MA dissertation, Alexandria University, Egypt, June 2010.

Analyzing Arabic Diacritization Errors Of MADA and Sakhr Diacritizer

Hamdy Mubarak, Ahmed Metwally, Mostafa Ramadan

Arabic NLP Researches, Sakhr Software
Sakhr Building, Free Zone, Nasr City 11711, Cairo
Egypt
{hamdys, amt, msr}@sakhr.com

Abstract

Modern standard Arabic (MSA) is usually written without diacritics, and this leads to morphological, syntactic, and semantic ambiguity. Diacritization (or diacritic restoration) is a very important basic step for several natural language processing (NLP) applications. In this paper, we present Sakhr Arabic disambiguation system that is used for selecting the best diacritization and sense for all words in Arabic text. We compare with the best performing reported system of Habash and Rambow (MADA) by analyzing errors in stem diacritization and case ending diacritization (using random samples from the GALE Dev10 newswire development data). We report the word error rate (WER) and diacritic error rate (DER) for both systems. Also, we give detailed statistics about different kinds of diacritization errors.

Keywords: Arabic NLP, Modern Standard Arabic (MSA), Arabic Diacritization, POS Disambiguation, Parsing

1. Introduction

Arabic is written with an orthography that includes optional diacritics typically representing short vowels. The absence of diacritics in modern standard Arabic (MSA) text is one of the most critical problems facing computer processing of Arabic text since this adds another layer of morphological and lexical ambiguity (one written word form can have several pronunciations, each pronunciation carrying its own meaning(s)).

Diacritization (*aka* vowelization, diacritic/vowel restoration) of Arabic text helps clarify the meaning of words and disambiguate any vague spellings or pronunciations. Diacritization is an important processing step for several natural language processing (NLP) applications, including part of speech (POS) disambiguation, training language models for Automatic Speech Recognition (ASR), Text-To-Speech (TTS) generation (Habash and Rambow 2007), in addition to Machine Translation (MT), and Arabic Data Mining applications (Shaalán et al., 2009).

Naturally occurring Arabic text has some percentage of diacritics, depending on genre and domain, to aid the reader disambiguate the text or simply to articulate it correctly. For instance, religious text such as the Holy Quran is fully diacritized to minimize the chances of reciting it incorrectly. Children's educational texts and classical poetry tend to be diacritized as well. However, news text and other genre are sparsely diacritized (e.g., around 1.5% of tokens in the United Nations Arabic corpus bear at least one diacritic) (Diab et al., 2007).

In this paper, we evaluate and analyze errors for two famous diacritization systems, namely the Morphological

Analysis and Disambiguation of Arabic (MADA) system (Habash and Rambow, 2005) and Sakhr Arabic Disambiguation System (ADS). The purpose is to highlight the most common errors in diacritization systems that need more focus and analysis to enhance accuracy.

This paper is organized as follows: Section 2 gives some examples and statistics about ambiguity in Arabic text due to lack of diacritics. Section 3 gives an overview about MADA. Section 4 describes Sakhr ADS. As for Section 5, it presents two experiments for evaluating these diacritization systems and detailed error analysis for each. Finally, section 6 gives some concluding remarks.

2. Ambiguity of Arabic Language

Arabic is a highly inflected language which has a rich and complex morphological system. MSA is very often written without diacritics, which leads to a highly ambiguous text. Arabic readers could differentiate between words having the same writing form (homographs) by the context of the script. For example, the word “علم Elm”¹ can be diacritized as “علم Eilm, science or knowing”, “علم Ealima, knew”, “علم Eallama, taught”, “علم Ealam, flag”, etc.

Debili, et al. (2002) calculate that an Arabic non-diacritized dictionary word form had 2.9 possible diacritized forms on average, and that an Arabic text containing 23K word forms showed an average ratio of 1:11.6 (quoted in Vergyri & Kirchoff 2004) (Maamouri et al., 2006).

¹ We use Buckwalter Arabic transliteration (Buckwalter, 2002) (<http://www.qamus.org/transliteration.htm>).

Maamouri and Bies (2010) show 21 different analyses of the Arabic word “ثمن” *vmn*, produced by BAMA. At SYSTRAN, which has been developing machine translation systems for over 40 years, it was estimated that the average number of ambiguities for a token in most languages was 2.3, whereas in MSA it reaches 19.2. Although ambiguity is caused primarily by the absence of short vowels, at SYSTRAN, researchers have found ambiguity in Arabic to be present at every level (Farghaly and Shaalan, 2009).

2.1 MSA Ambiguity in a POS-Tagged Corpus

For Sakhr POS-tagged corpus that contains 7M words gathered from different modern news services, we observed that MSA tends to be simpler than the Classical Arabic in grammar usage, syntax structure, morphological and semantic ambiguity. This helps normal Arabic readers to understand the written text easily. For example, 69% of words in this corpus have only 1 identified morphological analysis (one morphological interpretation), and 19% have 2 analyses, while high ambiguous words (3+ analyses) represent 12% only (Mubarak et al., 2009) as shown in Figure 1.

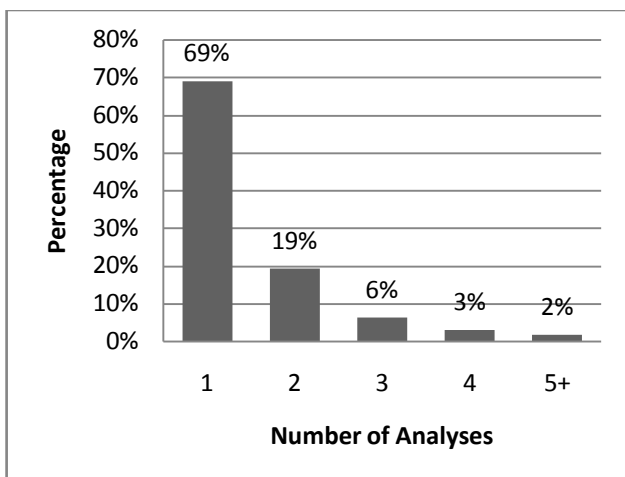


Figure 1: Distribution of Number of Word Analyses

Because Sakhr Morphological Analyzer provides an ordered list of analyses according to usage frequency, it was discovered that 92% of words occupy the first position in analyses, and 5% occupy the second one as shown in Figure 2, which means that MSA in most cases is not so ambiguous, and words occupy the “trivial” analysis! For example, the word “للحاكم” *liloHaAkmi*, to/of/for the ruler, “للحاكم” *liliHaAkumo*, to/of/for your beards, etc.), but the first one is usually recognized.

Figure 3 shows the distribution of case ending marks (mark on last letter) for nouns and verbs. We can observe that the case ending for verbs (if not given *غير مبني*) tends to be indicative (~81% of the cases), and for nouns (if not given) it tends to be genitive (~56% of the cases).

Figure 4 shows the distribution of diacritics extracted from the fully diacritized corpus. It is notable that “Fatha” is the most frequent diacritic and forms with “Kasra”, “Sukun” and “Damma” represent ~97% of the whole diacritics.

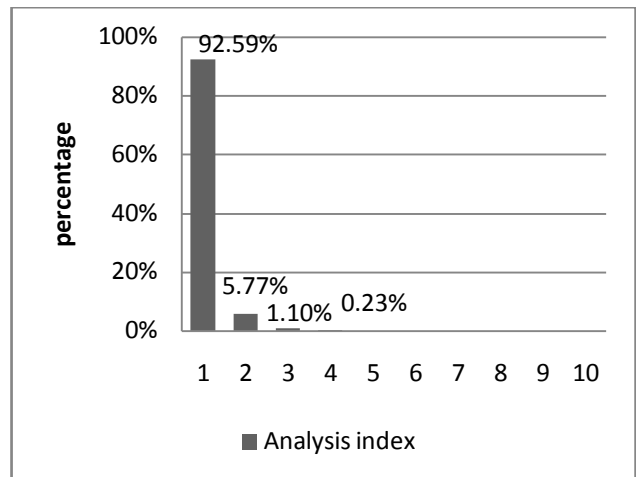


Figure 2: Distribution of the Selected Analysis Index

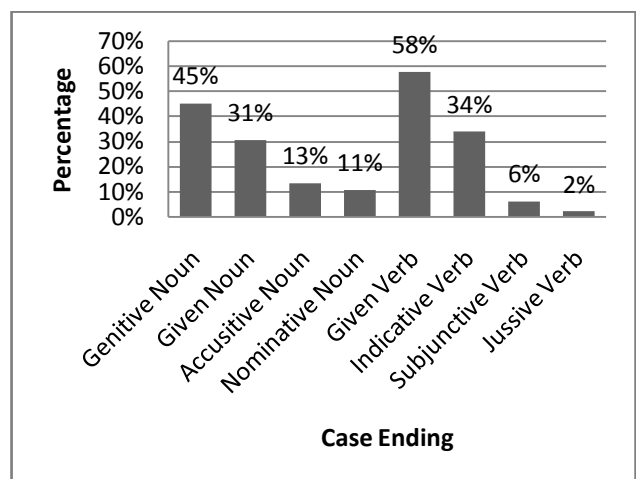


Figure 3: Case Ending Distribution

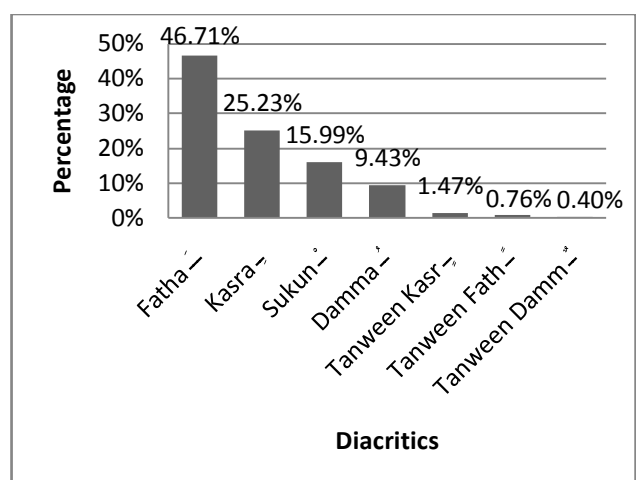


Figure 4: Diacritics Distribution

3. The MADA System

As mentioned in (Habash and Rambow, 2005), the basic approach used in MADA is inspired by the work of Hajic (2000) for tagging morphologically rich languages, which was extended to Arabic independently by Hajic et al. (2005). In this approach, a set of taggers are trained for individual linguistic features which are components of the full morphological tag (such as core part-of-speech, tense, number, and so on). In Arabic, we have ca. 2,000 to 20,000 morphological tags, depending on how we count. The Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004) is consulted to produce a list of possible analyses for a word. BAMA returns, given an undiacritized inflected word form, all possible morphological analyses, including full diacritization for each analysis. The results of the individual taggers are used to choose among these possible analyses. The algorithm proposed for choosing the best BAMA analysis simply counts the number of predicted values for the set of linguistic features in each candidate analysis.

Habash and Rambow (2007) introduced a system called MADA-D that uses Buckwalter's Arabic morphological analyzer where they used 14 taggers and a lexeme-based language model.

4. Sakhr Arabic Disambiguation System(ADS)

Sakhr morphological analyzer is a morphological analyzer-synthesizer that provides basic analyses of a single Arabic word, covering the whole range of modern and classical Arabic. For each analysis, it provides its morphological data such as diacritization, stem, root, morphological pattern, POS, prefixes, suffixes and also its morphosyntactic features like gender, number, person, case ending, etc. In addition to its high accuracy (99.8%), the morphological analyzer sorts the word analyses according to the usage frequency (using manual ordering of analyses for commonly-used words as appeared in an Arabic corpus of 4G words, or ordering according to stem frequency, otherwise). This morphological analyzer is integrated in most Sakhr products like TTS, MT, Search Engine and Text Mining.

ADS selects the best morphological analysis (which carries a large set of morphological data), and the best sense (which carries a large set of semantic data). Figure 5 is a screen shot that shows the diacritization for a random sentence¹.



Figure 5: ADS Diacritization

Figures 6-8 show the ADS morphological data (POS, diacritized stem, prefixes, suffixes, pattern, gender, number, person, etc), syntactic data (case ending, and attached pronoun), and semantic data (Arabic and English senses, semantic, ontological and thematic features).



Figure 6: ADS Morphological Disambiguation



Figure 7: ADS Syntactic Disambiguation

¹ ADS can be tested using website: <http://arabdiac.sakhr.com.eg>

program¹. These samples are diacritized using MADA² and Sakhr ADS.

We calculated errors manually for MADA and ADS considering **stem diacritization** (تشكيل البنية) and **case ending diacritization** (تشكيل الإعراب) for both samples³. We differentiate here between these errors as we believe that errors in stem diacritization are more important than errors in case ending diacritization for wide range of applications like TTS, MT, and text mining because this affects word meaning in most cases.

We found that number of stem diacritization errors for both samples for MADA was 141 (which represents 1.3%), and 108 (1.06%), while for ADS, the number was 35 (0.05%), and 32 (0.3%), and number of case ending diacritization errors for MADA was 509 (4.7%), and 400 (3.93%), while for ADS, the number was 222 (2.0%), and 180 (1.76%). Figure 10 shows these results.

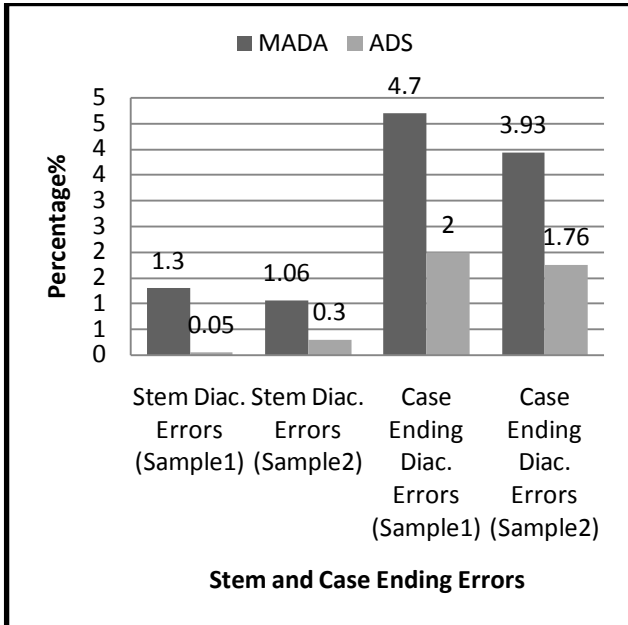


Figure 10: Stem and Case Ending Errors for MADA & ADS

5.1 Analyzing Stem Diacritization Errors

Error analysis for MADA shows that, on the average, 34% of stem diacritization errors are due to the lack of diacritics for unknown proper names, 30% are due to selecting wrong POS, and 16% are due to diacritizing some particles and function words incorrectly (namely, >n أن, <n إن, and mn من). The rest of errors (~20%) are mainly related to spelling mistakes and out of vocabulary (OOV) words. Figure 11 shows these errors in details and table 1 lists some examples for each type of errors.

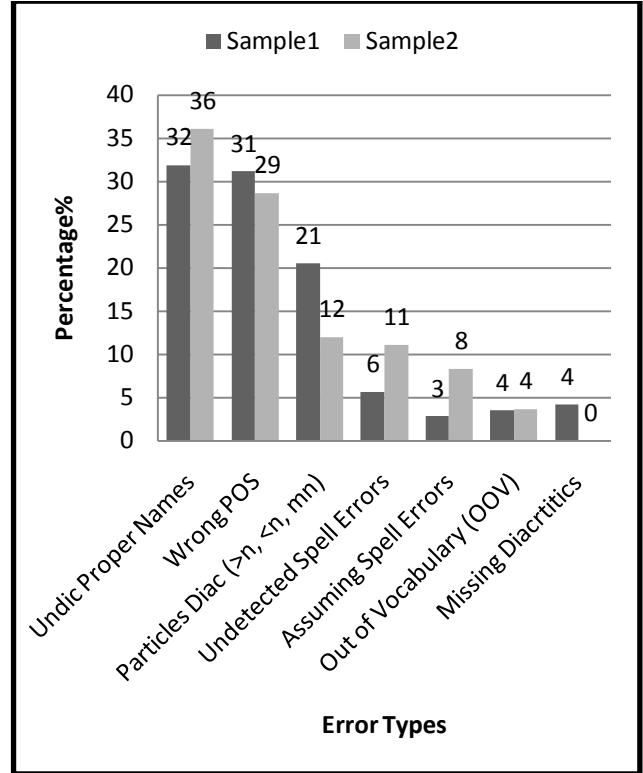


Figure 11: Error Analysis of Stem Diac. for MADA

MADA	
أمثلة	الخطأ
وَقَدْ التَّقَى سِرْكَيْسِيَانِ أَمْسُ كَاتُولِيكُوسِ الأَرْمَنِ / وَالتَّائِبُ أَعُوبُ بَقَرَادُونِيَانِ / تَقْرِيرِ غُولِدِسْتُونِ	أعلام غير مشكّلة
الإِتْفَاقِ التُّرْكِيِّ الأَرْمَنِيِّ المُرْمَعِ عَقْدَهُ / وَأَبْرَزَ قِيَادِيئِهَا فِي قِطَاعِ غَزَّةَ/ لَا أَشْعُرُ أَنِّي اسْتَحَقَّ أَنْ / مَعْرَبًا عَنِ أَمَلِهِ فِي	قسم كلم خاطئ
تَقُولُ أَنَّهُ لِأَعْرَاضِ سَلْمِيَّةٍ تَمَامًا / المَوْقِفِ النَّانِ هُوَ إِنْ مَبْدَأُ المُصَالِحَةِ قَائِمٌ / إِذْ أَنَّهُا لَنْ / بِأَنَّ يَطْلُبَ عَقْدَ إِجْتِمَاعِ عَاجِلٍ / أَنْ يَتَعَمَّدَ الفَقِيدَةَ / مَعَ كُلِّ مَنْ يَهْمُهُ الأَمْرُ / أَنْ مَنْ يَفْكَرُ فِي نَجَاحِ	تشكيل الأدوات: إن، أن، من
الصَّرَاغِ العَرَبِيِّ الإِسْرَائِيلِ / عِبْدَرَبِّهِ / مَا يُعَادَلُ غِطَاءَ 60.7 يَوْمٍ / مِنْ 61.4 يَوْمٍ فِي يُولْيُو / نِهَاجِيَةِ أَغْسُطُسِ (أب) / حَوْلَ بُنُودِ إِشْكَالِيَّةِ / أَوْقَفَ عَمِيَلَةَ البِنَاءِ /إِضَافَةَ إِلَى إِعْلَامِ فِلَسْطِينِ وَلُبْنَانَ	أخطاء إملائية لم يتم تصويبها
هُوَ وَأَهْمُ وَغَيْرِ واقِعِي / مِنْ جِهَةِ أُخْرَى هُنَا خَادِمِ الحَرَمَيْنِ / عَضُو المَجْلِسِ الوَطْنِيِّ الفِلَسْطِينِيِّ عَلَى قَيْصَلٍ / وَصَقَّهُ بِ "إِهَامِ جِدًّا"	تخطئة الكلمات الصحيحة
إِنِّي مِتْفَاجِي / مِتْرَافِقَةً مَعَ خُطَابِ سِيَاسِي / فِي عَضِيَّاتِ غَرَانِزِيَّةِ	كلمات خارج المعجم
لِشُؤُونِ الحُجَاجِ / تُحَدِيدِ سِنَّ الحُجَاجِ	نقص تشكيل

Table 1: Analysis of Stem Diac. Errors for MADA

On the other hand, error analysis for ADS shows that, on the average, 49% of stem diacritization errors are due to selecting wrong POS, 18% are due to undetected spelling errors, 16% are related to missing diacritics, and 12% are due to diacritizing some particles and function words incorrectly (namely, >n أن, <n إن, and mn من). The rest of errors (~5%) are mainly related to spelling mistakes (there

¹ <http://www ldc.upenn.edu/>

² We thank Nizar Habash for sharing MADA's output

³ If a word has any error in its stem diacritization, we count this as stem error, and if a word has any error in its case ending diacritization only, we count this as case ending error.

is no out of vocabulary (OOV) words). Figure 12 shows these errors in details and table 2 lists some examples for each type of errors.

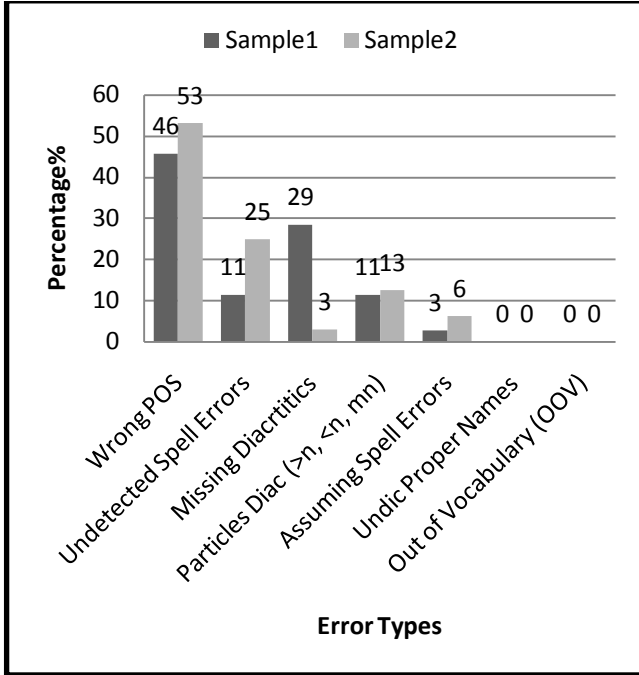


Figure 12: Error Analysis of Stem Diac. for ADS

ADS	
أمثلة	الخطأ
التفريير لا بُدَّ أن يُناقش في مجلس / فرار السلطة سحب تقرير / كل من تثبت إدانته	قسم كلم خاطئ
ما يُعادل غطاء 60.7 يوم / من 61.4 يوم في يوليو / أوقف عملية البناء / إضافة إلى إعلم فلسطين ولبنان	أخطاء إملائية لم يتم تصويبها
في تصريحات ل " الشرق الأوسط / بلال فرحات ل " الشرق الأوسط / مقارنة ب 61 مليون / يقبل الجائزة ك " نداء للعمل / إف 15 " و 14 طياراً	نقص تشكيل
الموقف الآن هو إن مبدأ المصالحة / قالت حركة المقاومة الإسلامية (حماس) أنه ما	تشكيل الأدوات: إن، أن، من
وتقديره للملك عبد الله علي النقة	تخنة الكلمات الصحيحة
لا يوجد	أعلام غير مشكّلة
لا يوجد	كلمات خارج المعجم

Table 2: Analysis of Stem Diac. Errors for ADS

5.2 Analyzing Case Ending Diac. Errors

Error analysis for MADA shows that, on the average, 28% of case ending diacritization errors are due to incorrectly recognizing subject and object, 15% are due to adjective relation, 14% are due to noun-noun relation "IDafa", 10% are due to conjunction relation, 7% of errors are due to prepositions attached to (or before) nouns, and 5% are due to subject and predicate recognition. The rest of errors (~21%) are mainly related to Inna and Kana sisters, adverbs, "tamyeez", etc. Figure 13 shows these errors in

details, and table 3 lists some examples for each type of errors.

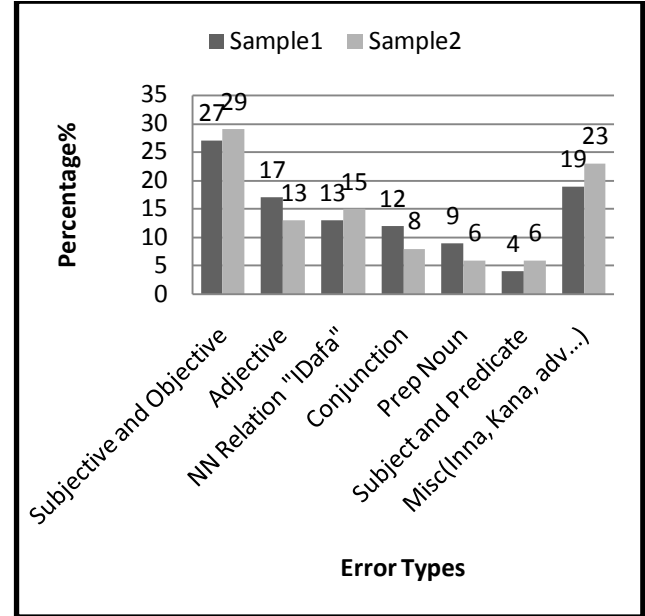


Figure 13: Error Analysis of Case Ending Diac. for MADA

MADA	
أمثلة	الخطأ
يُنَجُّ نظائر لعلاج / قدمتها دول مُفردة / خلقها هذه المبادرة الإجرامية / وتابع القول / طالب المكتب السياسي / أوضح المكتب السياسي / يتوجب محاسبة كل من تثبت / يتحكم إيقاع القمة التي جمعت العاهل السعودي	الفاعل والمفعول
مجلس الأمن الدولي / وأبلغ المجتمعون الرئيس الأرميني / التزام كامل / تعزيز مخزونات وفود / بموجة احتجاجات / وتوصيات تقرير / وكل جهات الاختصاص / لعلاج مرض السرطان	الصفة
بموجة احتجاجات واسعة واعتصامات قبالة استنقظ على مفاجأة / كتأكيد على القيادة الأبرككية / التواضع في تصريحه / كنداء للعمل / تزيد من العباء / وليس إلى أقوال " / من وباء إنفلونزا	الإضافة
الترجمة في لبنان لها حسابات / لها صدق في / هذه جائزة للمستقبل /	العطف
وأنه شخصياً مستمر في النضال / أنه مثل باقي الشعب / كانت هناك مقترحات / يمكن أن تكون أحد البائعين / حتى لا تكون هذه المصالحة شكلية	الجار والمجرور
	المبتدأ والخبر
	إن، كان، الظرف..

Table 3: Analysis of Case Ending Diac. Errors for MADA

On the other hand, error analysis for MADA shows that, on the average, 36% of case ending diacritization errors are due to incorrectly recognizing subject and object, 17% are due to adjective relation, 13% are due to conjunction relation, 10% are due to subject and predicate recognition, 7% are due to noun-noun relation "IDafa", and 3% are due to prepositions attached to (or before) nouns. The rest of errors (~14%) are mainly related to Inna and Kana sisters, adverbs, "tamyeez", etc. Figure 14 shows these errors in details and table 4 lists some examples for each type of errors.

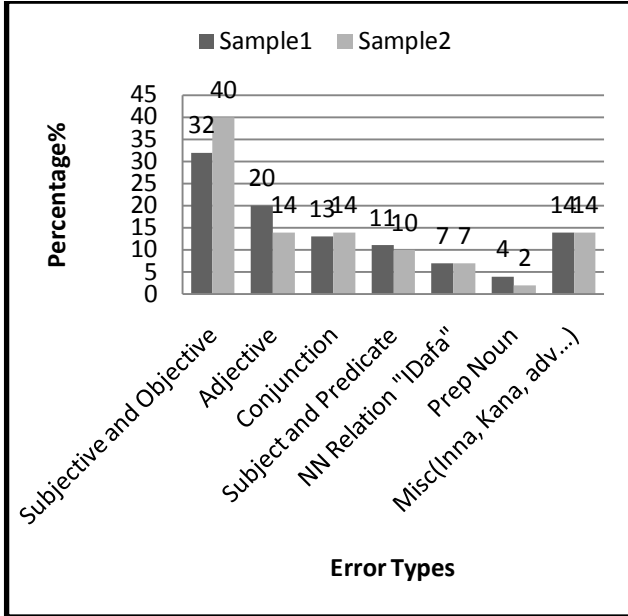


Figure 14: Error Analysis of Case Ending Diac. for ADS

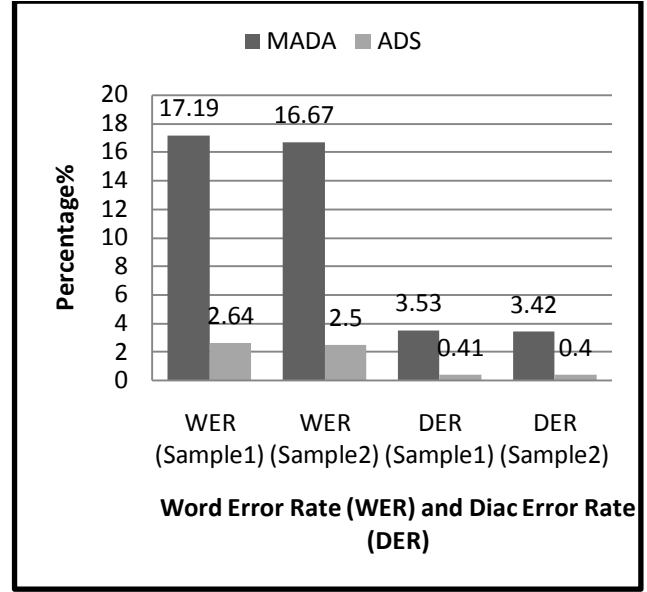


Figure 15: WER and DER for MADA and ADS

ADS	
أمثلة	الخطأ
وَتَابَعَ الْقَوْلَ / طَالِبَ الْمَكْتَبِ السِّيَاسِيِّ لِلجَبْهَةِ الشَّعْبِيَّةِ / يَتَوَجَّبُ مُحَاسِبَةً كُلَّ مَنْ تَنَبَّأَ / وَدَعَا الْمَكْتَبَ إِلَى ضَرْوَرَةٍ / سَيَبْتَطِلُ بِإِتِّحَادِ يَوْمِ / تَتَرَقَّبُ إِتْفِرَاجَاتِ تُؤَدِّي إِلَى /	الفاعل والمفعول
مِنَ الْوَلَايَاتِ الْمُتَّحِدَةِ عَنُوهَا الْقَدِيمِ / التُّرْكِيِّ - الْأَرْمَنِيِّ الْمَزْمُوعِ عَفْدَهُ / وَمُمْتَلِي الطَّوَائِفِ الْأَرْمِينِيَّةِ / الثَّلَاثِ /	الصفة
بِمَوْجَةِ إِحْتِجَاجَاتِ وَاسِعَةٍ وَأَعْتَصَامَاتِ / وَعَدَدٍ مِنْ الْفَعَالِيَّاتِ الْأَرْمِينِيَّةِ	العطف
الْمَوْقِفِ الْآنَ هُوَ / هُوَ وَاهِمٌ / سَوَاءٌ لَدَى خَلْفَانِهِ / لِبُنَانٍ جُزْءٍ مِنَ الْحَالَةِ الْإِقْلِيمِيَّةِ / لَهَا صَدَى فِي / هَذِهِ جَائِزَةٌ لِلْمُسْتَقْبَلِ	المبتدأ والخبر
مُحَاسِبَةً كُلَّ مَنْ / فَوْقَ رَأْسِ أَحَدٍ / مِنْ قِبَلِ بَعْضِ الْجِهَاتِ ،	الإضافة
ك " نِدَاءٌ لِلْعَمَلِ " / بِوَتِيرَةٍ أَسْرَعَ مِنْ التَّقْدِيرَاتِ / فِي مَسْعَى لِنَعْرِيزِ الرِّقَابَةِ / يَنْسَنُرُ عَلَى أَيِّ فَاسِدٍ أَنْ نَبَيْتَهَا الْحَقِيقِيَّةِ هِيَ بِنَاءٌ فَنَبْلَةُ نَوَوِيَّةِ / يُمْكِنُ أَنْ تُكُونَ أَحَدَ الْبَائِعِينَ/ سَيَكُونُ لَهُ نَتَائِجٌ / فِي زِيَارَةِ رَسْمِيَّةِ ، لِيَسْتَقْبِلَهُ / دَعُونِي أَكُونَ وَأَضِحًا / تَوْفَعَاتٍ إِقْتِصَادِيَّةٍ أَكْثَرَ نَقَاوِلًا / 252.6 مِليُونِ رِيَالٍ	الجار والمجرور إن، كان، الظرف..

Table 4: Analysis of Case Ending Diac. Errors for ADS

5.3 Calculating WER and DER

For the same samples, we calculated manually WER and DER for MADA and ADS. We found that MADA achieved an average WER of 16.93% and an average DER of 3.4% compared to ADS which achieved a WER of 2.57% and a DER of 0.4%. This is shown in Figure 15.

It is observed that MADA has common problems that can be easily enhanced to minimize both WER and DER. These problems can be classified as a missing diacritic in the following cases:

- "moon Lam القمرية الإيرانية" (ex: الإيراني Al<irAniy~)
- letters before vowels (ex: مخمود maHomwd).
- last letter in function words with/out suffixes (ex: من min, عنه Eanhu)
- last letter of some suffixes(ex: حقوقهم Huqukihim)
- "feminine Taa التانيث المفتوحة" (ex: عرضت Earadat)

The following figure shows these missing and wrong diacritics for MADA and ADS for an arbitrary sentence.

MADA
قال الرئيس الإيراني، محمود أحمددي نجاد، إن بعض الدول عرضت تزويد بلاده بـ 20 في المائة لاستخدامه كوقود نووي.
ADS
قال الرئيس الإيراني، محمود أحمددي نجاد، إن بعض الدول عرضت تزويد بلاده بـ 20 في المائة لاستخدامه كوقود نووي.

Figure 16: Highlighting Diacritization Errors

Because there is no standard test bench for measuring WER and DER, we just summarize in the following table some reported evaluation experiments for different diacritizers.

Engine	MADA	Zitouni	Sakhr ADS	RDI	Shaalán	KACST
Evaluator						
MADA (Habash, N.)	14.9 4.8					
Zitouni (Zitouni, I.)		18.0 5.5				
Sakhr ADS (Mubarak, H.)	16.9 3.4		2.6 0.4			
RDI (Rashwan, M.)	14.9 5.5	18.0 7.9		12.5 3.1		
Shaalán (Shaalán, K.)					11.8 3.2	
KACST (Alghamdi, M.)						26.0 9.2

Table 5: WER% and DER% (in order) for some diacritizers

6. Conclusions

In this paper, we presented Sakhr Arabic disambiguation system (ADS) which resolves morphological, lexical, and semantic ambiguity in Arabic texts. We compared the ADS diacritization with the best diacritization system that is reported in the literature so far (MADA). We analyzed errors in diacritizing stem and case ending for both engines, and measured word error rate (WER) and diacritic error rate (DER). We recommend here to have a standard test bench for evaluating different Arabic diacritizers, and also to measure both stem errors and case ending errors separately as their impacts on word meaning are not the same.

7. References

- [1] Alghamdi, M., Muzaffar, Z. (2007). KACST Arabic Diacritizer. The First International Symposium on Computers and Arabic Language.
- [2] Diab, M., Ghoneim, M., and Habash, N. (2007). Arabic Diacritization in the Context of Statistical Machine Translation, MT Summit XI, Copenhagen, Denmark.
- [3] Elshafei, M., Almuhtasib, H., and Alghamdi, M. (2006). Machine Generation of Arabic Diacritical Marks. The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing. Las Vegas, USA.
- [4] Farghaly, A. and Shaalan, K. (2009). Arabic Natural Language Processing: Challenges and Solutions. ACM Transactions on Asian Language Information
- [5] Habash, N., and Rambow, O. (2007), Arabic Diacritization Through Full Morphological Tagging, The North American Chapter of the Association for Computational Linguistics (NAACL).Rochester, New

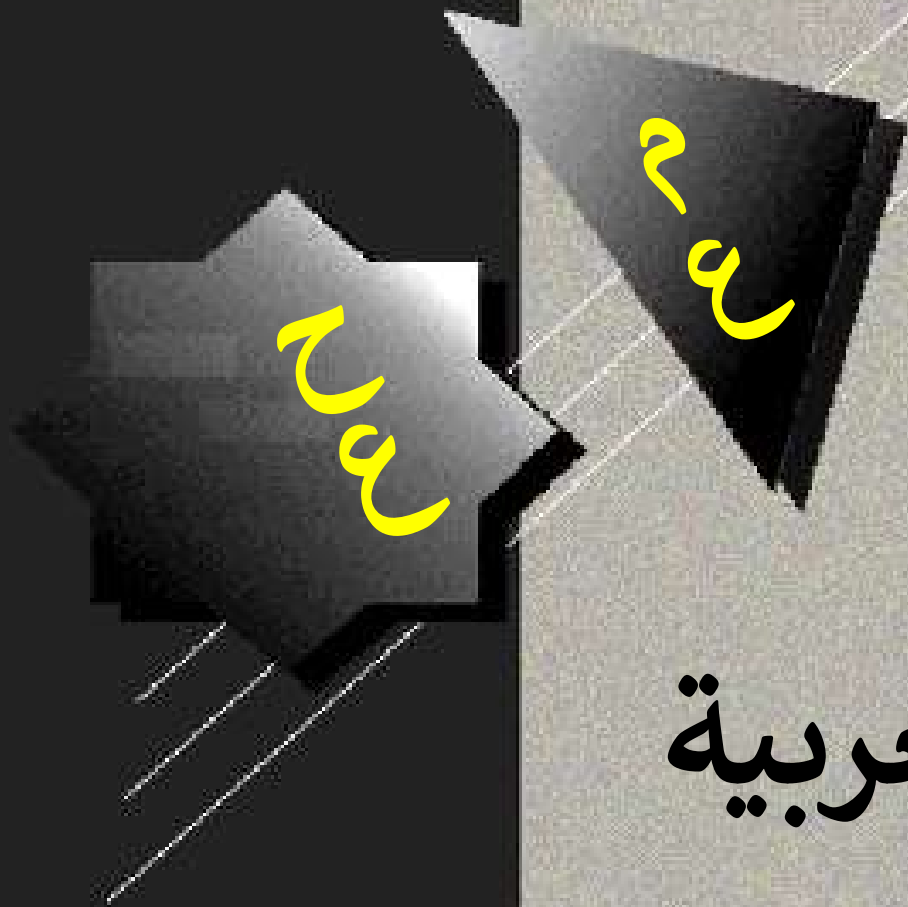
York.

- [6] Maamouri, M., Bies, A., and Kulick, S. (2006). Diacritization: A Challenge to Arabic Treebank Annotation and Parsing. In Proceedings of the Conference of the Machine Translation SIG of the British Computer Society.
- [7] Mubarak, H., Shaban, K., and Adel, F. (2009). Lexical and Morphological Statistics of an Arabic POS-Tagged Corpus. The 9th Conference on Language Engineering, Cairo, Egypt.
- [8] Rashwan, M., Al-Badrashiny, M., Attia, M., Abdou, S, Rafea, A. (2011) . A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features, IEEE Transactions on Audio, Speech, and Language Processing.
- [9] Shaalan, K., Abo Bakr, H., Ziedan, I. (2009). A Hybrid Approach for Building Arabic Diacritizer, Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, Association for Computational Linguistics. Athens, Greece.
- [10] Zitouni, I., Sorensen, J. S., and Sarikaya, R. (2006). Maximum Entropy Based Restoration of Arabic Diacritics, in Proceedings of ACL'06.

جمعية هندسة اللغة

جامعة عين شمس - القاهرة

14 - 15 ديسمبر 2011م



حوسبة العربية ومجتمع المعرفة

د. نبيل علي

الرسالة من البدايئة

انفجار المعرفة = انفجار الطلب على هندسة اللغة

كيفية البحث عن مناهل البحوث والتطوير؟

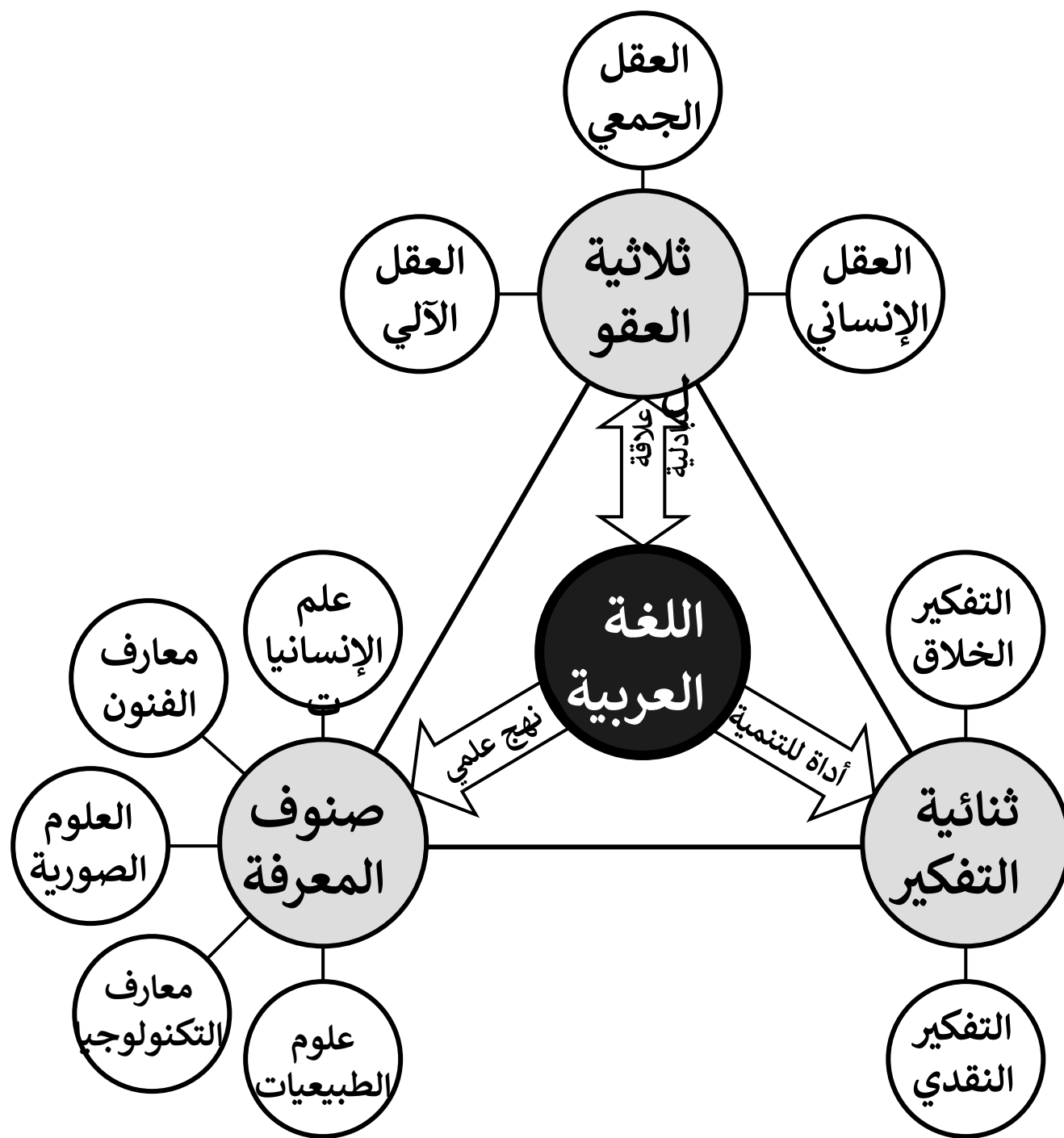
الإطار العام

1 • اللغة في موقع القلب

2 • مدخل العقول

3 • مدخل أطوار التفكير

4 • مدخل صنوف المعرفة



T.S. ELLIOT'S DISAPPOINTMENT:

LOSS OF WISDOM IN THE KNOWLEDGE MAZE

TIM BURNER LEE'S DISSAPPOINTMENT:

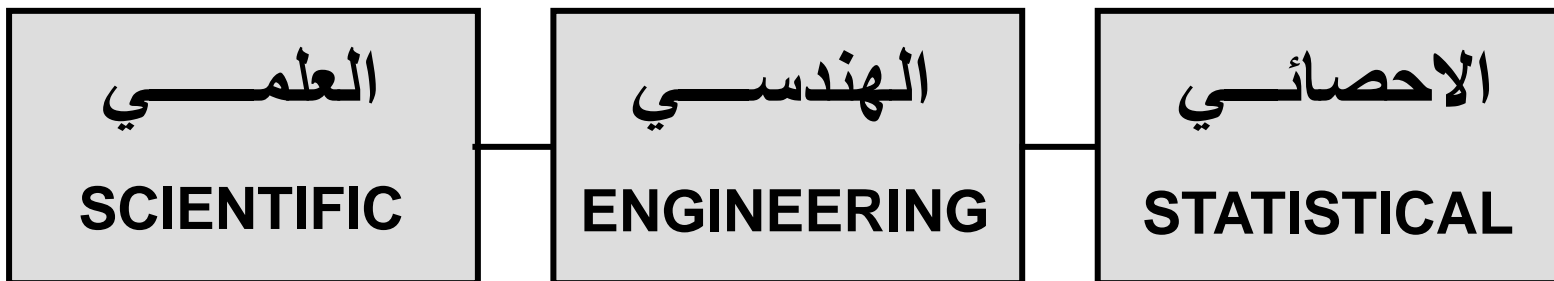
LOSS OF KNOWLEDGE IN THE INFORMATION MAZE

CONFRONTING THE INFOAMTION OVEREAD:

UNFAIR DIVISION OF LABOR

المفهوم

- المفهوم هو وحدة بناء المعرفة
- المفهوم ما قبل وبعد لغوي PRE/ POST LINGUISTIC
- تمدد المفهوم وتتعدد العبارات الدالة عليه
- لكل مجال معرفي بنيته المفهومية (الانطولوجيا)



نحن هنا

الويب الدلالي ما قبل الذكاء الاصطناعي SW: PRE - AI



نتعني لكم بوابات البحث التقليدية

SEARCH PORTALS ARA MORTAR

MULTIFACETED COMPUTATIONAL LINGUISTICS (CL)

	C	L	
CONTENT	C	L	LINGUISTICS
CONTEXT	C	L	LOGIC
CONCEPT	C	L	LEXICON

التدقيق الدلالي : من وحي الساعة

إحصائيا	برجماتيا	دلاليا	مقطع النص
NO		NO	ريادة المـوارد
NO		NO	التفكير الإبداعي
NO	NO		التفكير والهجرة
NO			ثروة الجوع
NO		OK	ثروة الجوع التي حصلوا عليها
	NO	OK	سرقة ثورة شباب 25 يناير
NO		OK	ثروة السباب

العمومية

UNIVERSALITY

الويب
الدلالي

U

M

تعدد اشكال
التنظيم

MULTIFORMAT

التوحيد

UNIFICATION

SEMANTIC
WEB

التشكيل
الميكروي

MICROFORMAT

ثنائيات حاكمة في التعامل مع النصوص

استنتاج INFERENCING	استدلال	تمثيل	أشياء - أحداث THINGS-EVENTS
استفهام QUERY	اتوماتي	المعرفة	مفاهيم CONCEPTS
برهنة PROOF	AUTOMATIC REASONING	KNOWLEDGE REPRESENTATION	علاقات RELATIONS
لسانيات نصية TEXT LINGUISTICS	طبيعي	صوري	منطق LOGIC
لسانيات نصوص CORPUS LINGUISTICS	NATURAL	FORMAL	رياضي MATHEMATICAL

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

K	KE	KM	KA	KD	KR
----------	-----------	-----------	-----------	-----------	-----------

K	KE	KM	KA	KD	KR
KNOWLEDGE المعرفة	هندسة المعرفة	إدارة المعرفة	اقتناء المعرفة	اكتشاف المعرفة	تمثيل المعرفة

META	META ATA	META COG	META COM	META SEM	META TEXT
-------------	---------------------	---------------------	---------------------	---------------------	----------------------

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL

L	PL	PPL	ILP	SL	DL
LOGIC منطق	منطق مقولات	منطق اسنادي	معالجة النحو الاستقرائي	منطق الاحتشاد	منطق وصفي

ATA CCG COM SEM TEXT

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW

O	OE	OWL	OIL	OM	SW
ONTOLOGY انطولوجيا	هندسة انطولوجية	لغة الويب الانطولوجية	لغة الاستنتاج الانطولوجي	إدارة الانطولوجيات	ويب داللي

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

D	DM	TDM	KDB	DD	SN
DATA البيانات	التنقيب في ذخائر البيانات	التنقيب في ذخائر النصوص	قاعدة بيانات	قاموس بيانات	شبكة دلالية

D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

EMERGENCE

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

M	ML	MR	MT	IM	SS
MACHINE آلة	تعلم الآلة	تمييز آلي	ترجمة آلية	آلة استنتاج	تركيب دلالي

M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

K	KE	KM	KA	KD	KR

META	META ATA	META COG	META COM	META SEM	META TEXT
DETA ميٽا	ميٽا بيانات	ميٽا معرفي	ميٽا تواصلتي	ميٽا دلالي	ميٽا نصبي

META	META ATA	META COG	META COM	META SEM	META TEXT
-------------	---------------------	---------------------	---------------------	---------------------	----------------------

EMERGENCY

انبثاق

K	KE	KM	KA	KD	KR
L	PL	PPL	ILP	SL	DL
O	OE	OWL	OIL	OM	SW
D	DM	TDM	KDT	DD	SN
M	ML	MR	MT	IM	SS
META	META ATA	META COG	META COM	META SEM	META TEXT

ما
الذي
يجري
من
حولنا
!!?..

مقدمة الرأس



حضانة النصف الأيمن



	جوانب اللاقطعية												
	1	2	3	4	5	6	7	8	9	10	11	12	
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
القدرة على التعميم			●	●		●	●	●	●	●	●	●	●
القدرة على التنسيق								●					
الاستخلاص من المشوش		●						●					●
إكمال الناقص		●	●				●	●					●
استئناس غير الدقيق		●	●					●					●
الصمود إزاء التعقد	●	●	●	●	●	●	●	●	●	●	●	●	●
تنوع تنمية القدرات الذهنية	●						●	●	●	●	●		

فض البس

الذف

الدلالة اللغوية

الدلالة المعجمية

الإحالة، الإضمار وخلافه

اللغويات الإحصائية

	1	2	3	4	5	6	7	8	9	10	11	12
جوانب اللاقطعية												
القدرة على التعميم			•	•		•	•	•	•			
القدرة على التنسيق						•						
الاستخلاص من المشوش			•			•					•	
إكمال الناقص			•	•		•	•					
استئناس غير الدقيق			•	•			•					
الصمود إزاء التعقد	•	•	•	•	•	•	•	•	•	•	•	•
تنوع تنمية القدرات الذهنية	•					•	•	•	•			

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	جوانب اللاقطعية
1															القدرة على التعميم
2															القدرة على التنسيق
3															الاستخلاص من المشوش
4															إكمال الناقص
5															استئناس غير الدقيق
6															الصمود إزاء التعقد
7															تنوع تنمية القدرات الذهنية
8															
9															
10															
11															
12															
13															
14															

14 الواقِع الخائلي

13 الروبوتات المعرفية

12 الهندسة الانطولوجية

11 الويب الدلالي

10 الحوسبة الكوانتية

9 الاستخلاص والتخليص الآلي

8 الفهرسة على أساس الكلمات والمفاهيم

7 محركات البحث الذكي

6 الوسائط المتعددة

5 معالجة اللغات الطبيعية

4 الذكاء الاصطناعي

3 شبكات الاتصالات

2 النمذجة والمحاكاة

1 نظرية المعلومات

اللسانيات الحاسوبية ذات الوجهين

نظرية المعلومات
النمذجة والمحاكاة
الذكاء الاصطناعي
معالجة اللغات الطبيعية
محركات البحث الذكي
الويب الدلالي

حاسوبي لساني

استكمال
النقص

لساني حاسوبي

فض الـبـس
الـحـذف
الدلالة اللغوية
الدلالة المعجمية
الإحالة، الإضمار وخلافه
اللغويات الإحصائية



التقييم

6

يؤول يصدر حكما يتحقق من ينقد

يحل يقرر يبرر يعترض يخلص إلى

التركيب

5

يضيف يدمج يترجم يتنبأ يؤالف

يوسع يفترض يصمم يعيد بناء يعدل

التحليل

4

يقارن يقابل يستتبط يحدد يجنب

يربط يفكك يحدد يقرر يفرق

المستويات
العليا

التطبيق

3

ينظم يجمع يصنف يقيني يطبق يلخص

يرتب يستخدم ينمذج ييني يربط يكود

المستويات
الدنيا

الفهم

2

يفسر

يلخص

يقيم الصلة

يعيد تجميع

يعيد صياغة

إعادة الصياغة الأسلوبية الحاسوبية

AUTOMATIC PARAPHRASING
COMPUT. STYLISTICS

المعرفة

1

يشرح

يقرر

يصف

يتذكر

التقييم

6

يؤول يصدر حكما يتحقق من ينقد

يحل يقرر يبرر يعترض يخلص إلى

التركيب

5

يضيف يدمج يترجم يتنبأ يؤالف

يوسع يفترض يصمم يعيد بناء يعدل

التحليل

4

يقارن يقابل يستتبط يحدد يجنب

يربط يفكك يحدد يقرر يفرق

المستويات
العليا

يصنف

3

التطبيق

يلخص يلق

يكود يبط

ينظ

يرت

المستويات
الدنيا

نظم التصنيف الاتوماتي

AUT. CLASSIFICATION
SYSTEMS

يقيم الصلة

يعيد تجميع

يفسر

يلخص

الفهم

2

يشرح يسرد يلاحظ يؤول يصوغ يصف

يقرر يجرب يكشف يميز يخبر يتذكر

المعرفة

1

التقييم

6

يؤول يصدر حكما يتحقق من ينقد

يحل يقرر يبرر يعترض يخلص إلى

التركيب

5

يضيف يدمج يترجم يتنبأ يؤالف

يوسع يفترض يصمم يعيد بناء يعدل

التحليل

4

يقارن
يربط

يستنبط

يجنب
يفرق

آلة الاستنباط العربية

المستويات
العليا

المستويات
الدنيا

التطبيق

3

ينظم
يرتب

يلخص
يكود

ARABIC INFERENCE M/C

الفهم

2

يفسر يترجم يعيد صياغة يصوب يقيم الصلة
يلخص يصف يعرض يعيد تجميع

المعرفة

1

يشرح يسرد يلاحظ يؤول يصوغ يصف
يقرر يجرب يكشف يميز يخبر يتذكر

المعرفة

يقرر

يجرب

يكشف

يميز

يخبر

يتذكر

يشرح

يسرد

يلاحظ

يوؤل

يصوغ

يصف

يلخص

يصف

يعرض

يجمع

يعيد تجميع

الفهم

يفسر

يترجم

يعيد صياغة

يصوب

يقيم الصلة

التطبيق

يرتب

يستخدم

ينمذج

يبنى

يربط

يكود

ينظم

يجمع

يصنف

يقيني

يطبق

يلخص

المستويات
الدنيا

التحليل

يقارن

يربط

التحليل

التركيب

يضيئ

يوسد

التقييم

يوؤل

يصدر حكما

يتحقق من

ينقد

يحل

يقرر

يبزر

يعترض

يخلص إلى

يؤلف

يعدل

ينب

يرق

نظم الترجمة والتحويل الآلية

AUT. TRANSLATION &
TRANSFORMATION

يترجم

يُتْحَقَّقُ مِنْ

التماسك السياقي والترابط المنطقي

LOG. COHERENCE &
CONTEXT. COHESION

التقييم

6

التركيب

5

التحليل

4

التطبيق

3

الفهم

2

المعرفة

1

يقارن يقابل يستتبط يحدد يجنب
يربط يفكك يحدد يقرر يفرق

ينظم يجمع يصنف يقيني يطبق يلخص
يرتب يستخدم ينمذج يبني يربط يكود

يفسر يترجم يعيد صياغة يصوب يقيم الصلة
يلخص يصف يعرض يعيد تجميع

يشرح يسرد يلاحظ يؤول يصوغ يصف
يقرر يجرب يكشف يميز يخبر يتذكر

المستويات
العليا

المستويات
الدنيا

مجتمع
المعرفة

الثقافة

التربية

اللغة

الإبداع

الإعلام

الاقتصاد

علم النفس المعرفي

علم النفس الثقافي

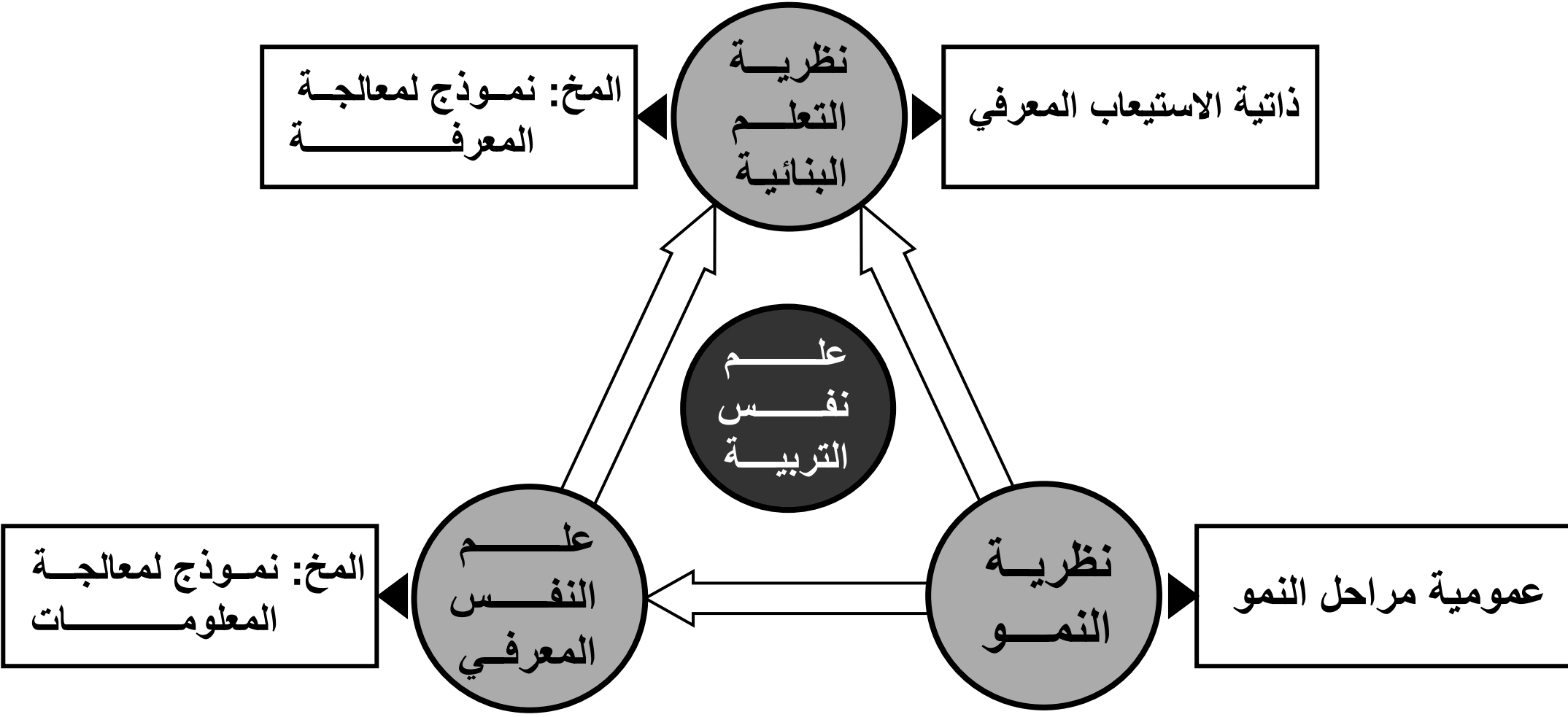
علم النفس التربوي

علم النفس اللغوي

علم نفس الإبداع

اقتصاد الانتباه

علم
النفس



From Data to Nuanced Information Making Implicit Knowledge Explicitly Useful

Mona Diab
Columbia University

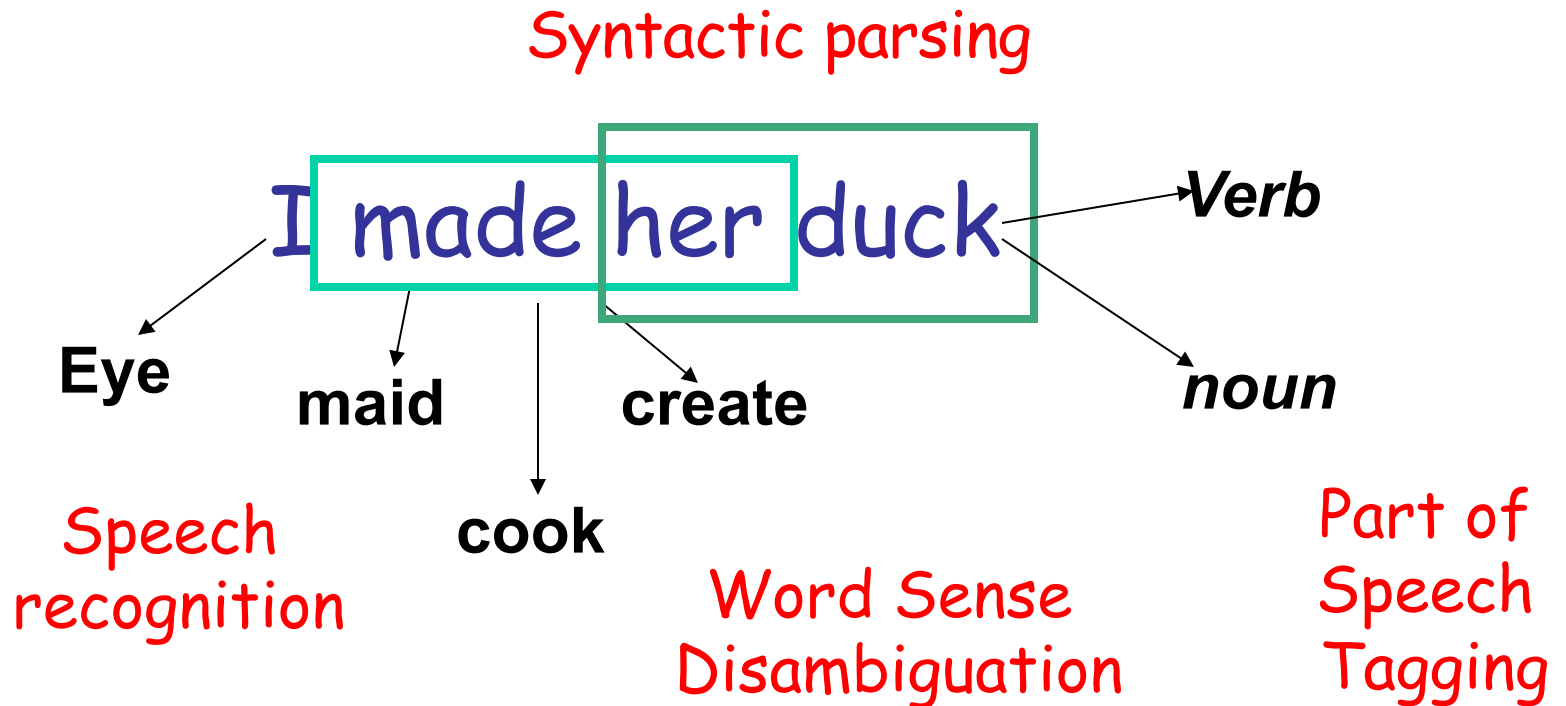
Automatic Language Processing

The Challenge is Ambiguity

I made her duck

- I cooked waterfowl for her
- I cooked the waterfowl that belongs to her
- I created the ceramic duck she owns
- I caused her to quickly lower her head
- *And more....*

Pervasive Ambiguity



Ambiguity Resolution

- Ambiguity results from the existence of multiple possibilities at each level
 - All levels of linguistic knowledge require resolving ambiguity
- Solution
 - Divide & Conquer
 - Optimization & Constraint Satisfaction
 - Efficient Search

Why NLP?

- kJfmmfj mmmvvv nnnffn333
- Uj iheale elee mnster vensi credur
- Baboi oi cestnitze
- Coovoel2^ ekk; ldsllk lkdf vnnjfj?
- Fgmflmlk mlfm kfre **xnnn!**
- *Can you READ this? You, yes you!*

Computers Lack Knowledge

- Computers “see” text in English/Arabic the same way you saw the previous slide!
- **People** have no trouble understanding language
 - Common sense knowledge
 - Reasoning capacity
 - Experience
- However, **Computers** have
 - No common sense knowledge
 - No reasoning capacity

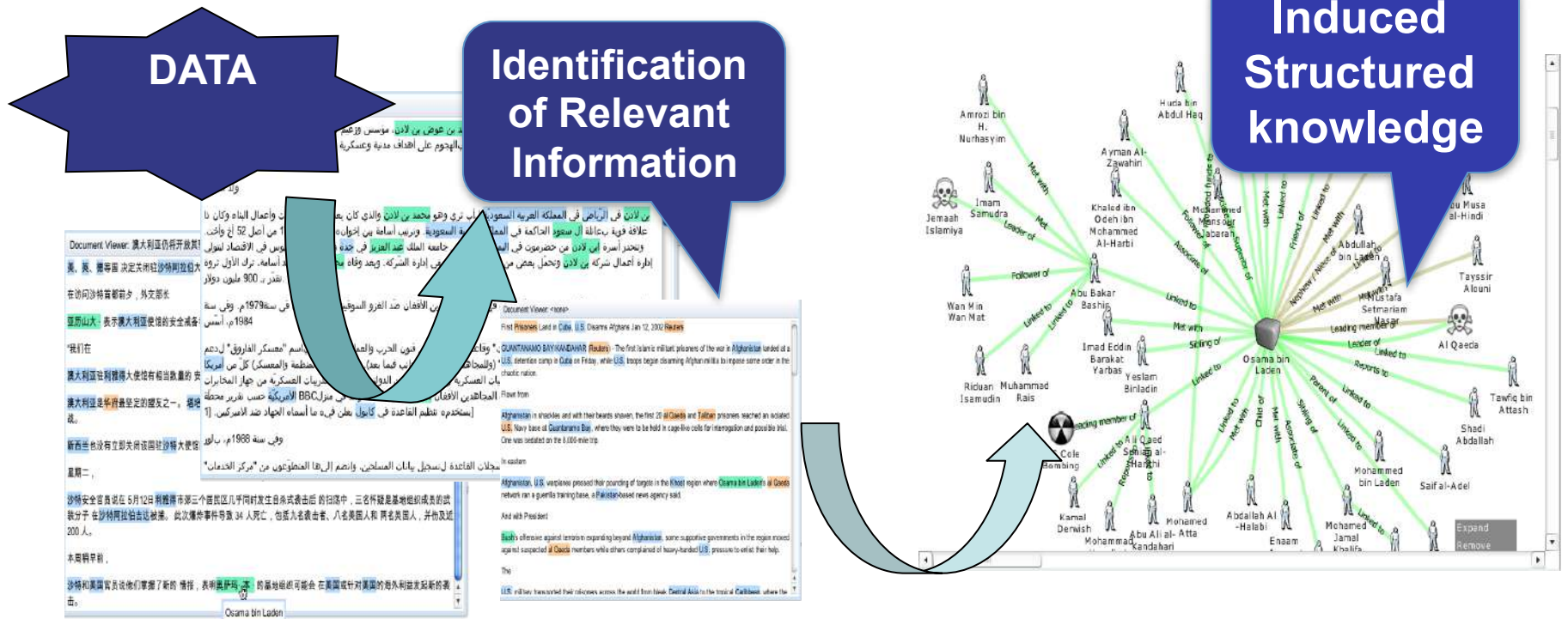
Unless we teach them!

One CL/NLP Objective

- Take people's everyday language (the way they speak/write) and do useful things with it such as:
 - Translate from one language to another
 - Extract relevant information for a task (distillation, summarization, track opinions, gage people's sentiments towards something/someone)
 - Information retrieval (google/yahoo/bing)
 - Improve pedagogical systems
 - etc....

Example Information Extraction

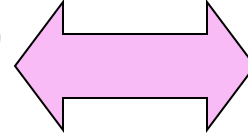
- Robustly handling/processing of meaning in context for different applications



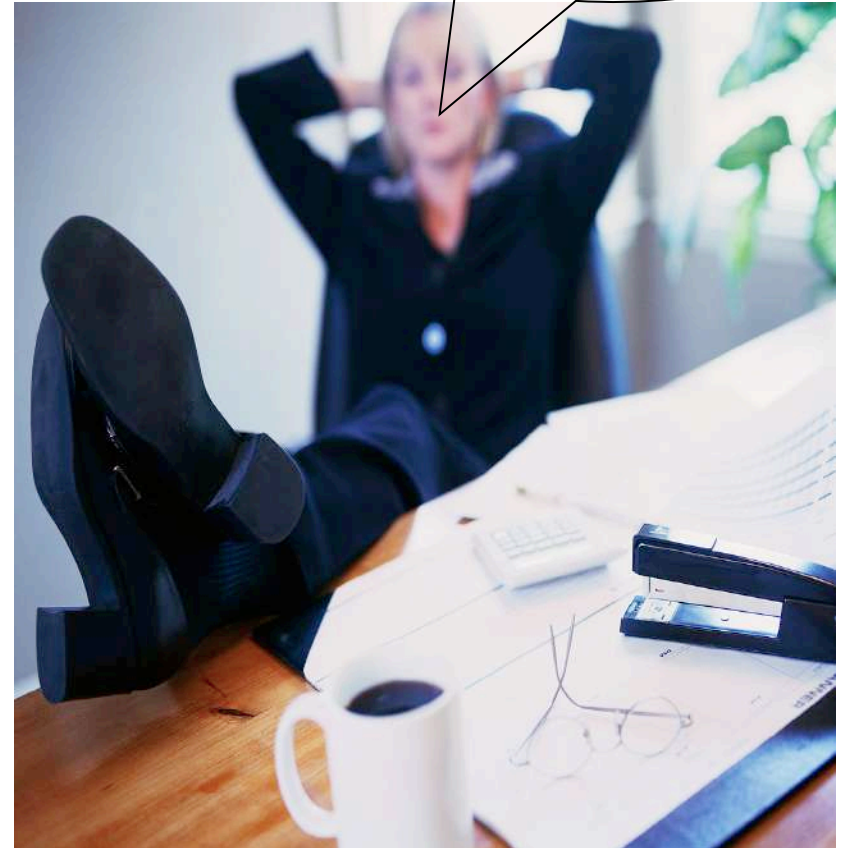
For Question Answering

ايه رايكوفي
قلقاس

01101
00101
0110



هناكل ايه
النهارده ياتارا؟



Two Relevant Enabling Technologies for Information Extraction

- Named Entity Recognition (NER): for answering questions such as "*Who killed Kennedy?*" or "*Where was Obama born?*"
- Semantic Role Labeling (SRL): for identifying *who did what to whom when, how and why.*

RoadMap of Talk

- SRL
 - How it is tailored for Arabic
 - Extending Tree Kernels
- NER
 - How is it tailored for Arabic and different from English
 - A subclass of Multiword Expressions
 - Integration considerations in Machine Translation

What is SRL?

Proposition

John opened the door

What is SRL?

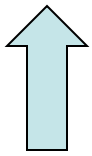
Proposition

[John]_{Agent} [opened]_{Predicate} [the door]_{Theme}

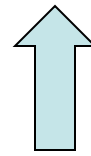
What is SRL?

Proposition

[John]_{Agent} [opened]_{Predicate} [the door]_{Theme}



Subject

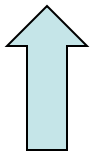


Object

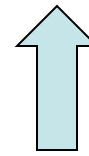
What is SRL?

Proposition

[John]_{Agent} [opened]_{Predicate} [the door]_{Theme}



Subject



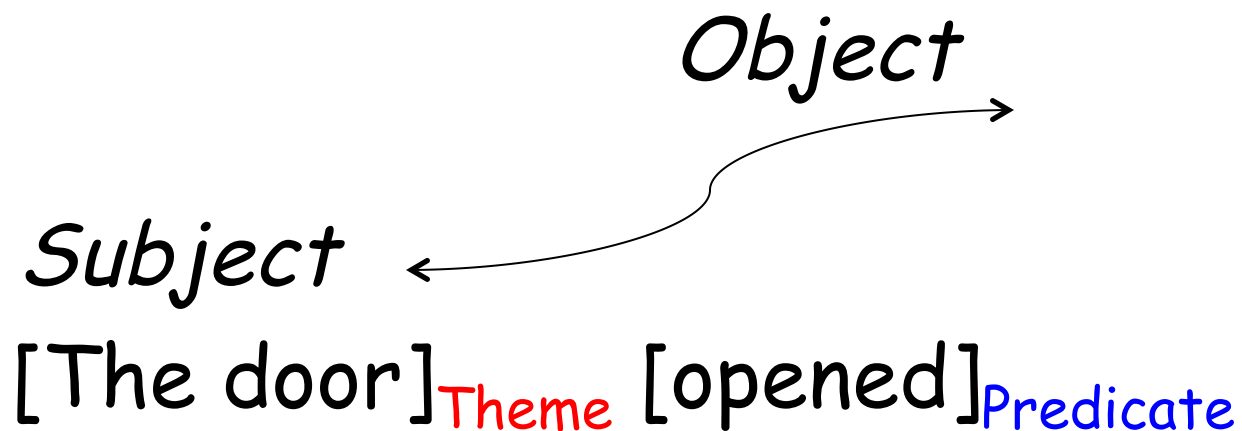
Object

[The door]_{Theme} [opened]_{Predicate}

What is SRL?

Proposition

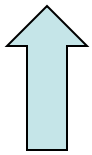
[John]_{Agent} [opened]_{Predicate} [the door]_{Theme}



What is SRL?

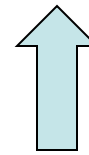
Proposition

[John]_{Agent} [opened]_{Predicate} [the door]_{Theme}

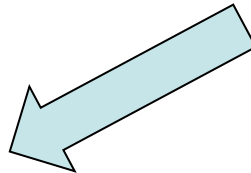


Agent

FrameNet



Container_portal

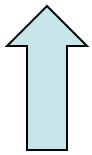


[The door]_{Theme} [opened]_{Predicate}

What is SRL?

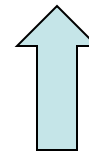
Proposition

[John]_{Agent} [opened]_{Predicate} [the door]_{Theme}

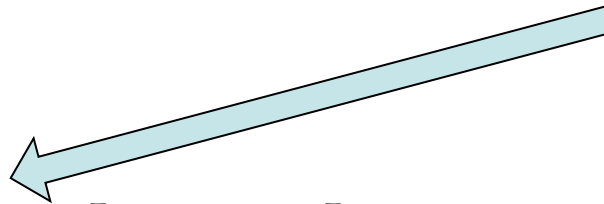


ARGO

PropBank



ARG1



[The door]_{Theme} [opened]_{Predicate}

Our Goal

بدأ رئيس الوزراء الصيني زو رونغجي زيارة رسمية لالهند الاحد الماضي



Last Sunday India to official visit Rongji Zhu the-Chinese the-Ministers president started

The Chinese Prime Minister Zho Rongji started an official visit to India last sunday

Our Goal

ARGM-TMP

ARG1

ARG0

بدأ [رئيس الوزراء الصيني زو رونغجي] [زيارة رسمية ل الهند] [الاحد الماضي]



Last Sunday India to official visit Rongji Zhu the-
Chinese the-Ministers president started

The Chinese Prime Minister Zho Rongji started an
official visit to India last Sunday

Arabic(s)

- Arabic is a Semitic language
- Forms of Arabic
 - Classical Arabic (CA)
 - Classical Historical texts
 - Liturgical texts
 - Modern Standard Arabic (MSA)
 - News media & formal speeches and settings
 - Only written standard
 - Dialectal Arabic (DA)
 - Predominantly spoken vernaculars
 - No written standards
- Dialect vs. Language
 - Linguistics vs. Politics

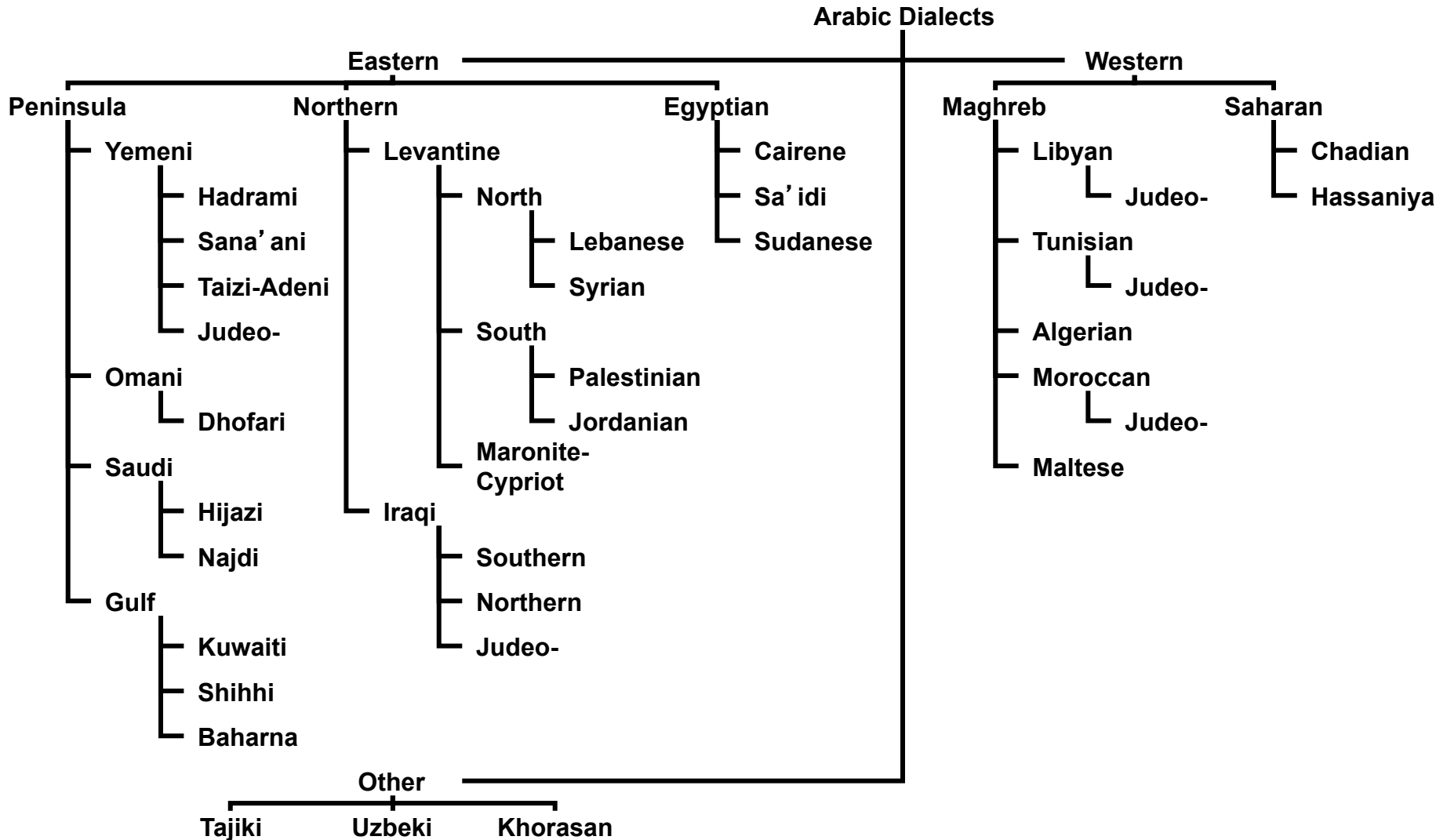
Introduction

- ~300M people worldwide speak Arabic
- Arabic is **the**/an official language of 23 countries
- No native speakers of CA nor MSA
- In the Arabic speaking world, MSA and CA are the only Arabic taught in schools

Introduction

- Arabic Diglossia
 - Diglossia is where two forms of the language exist side by side
 - MSA is the formal public language
 - Perceived as “language of the mind”
 - Dialectal Arabic is the informal private language
 - Perceived as “language of the heart”
- General Arab perception: dialects are a deteriorated form of Classical Arabic
- Continuum of dialects

Geographical Continuum



Social Continuum

- Factors affecting dialect
 - Lifestyle
 - Bedouin, urban, rural
 - Education & Social Class
 - Religion
 - Muslim, Christian, Jewish, Druze, etc.
 - Gender

Arabic & its Dialects

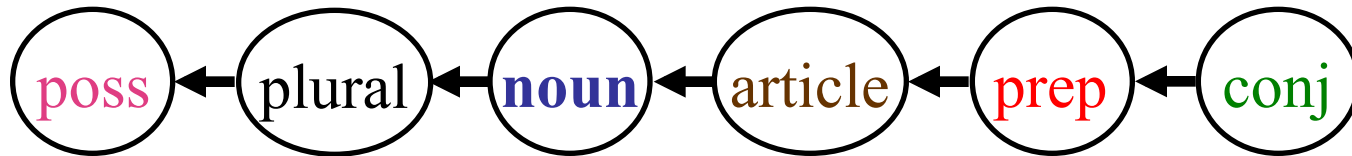
- Degrees of linguistic distance

	Syntax	Morphology	Lexicon	Phonology
MSA-Dialect	++	+++	++++	++++
Inter-Dialect	+	+++	++++	++++
Intra-Dialect	0	0	+	+

- Lack of standards for the dialects
- Lack of written resources

Inflectional Morphology

Nouns



وأكبيوتنا

/wakabiyūtinā/

نا + بيوت + ك + و

wa+ka+biyūt+nā

and+like+houses+our

And like our houses

وللمكتبات

/walilmaktabāt/

و+ل+ال+مكتبة+ات

wa+li+al+maktaba+āt

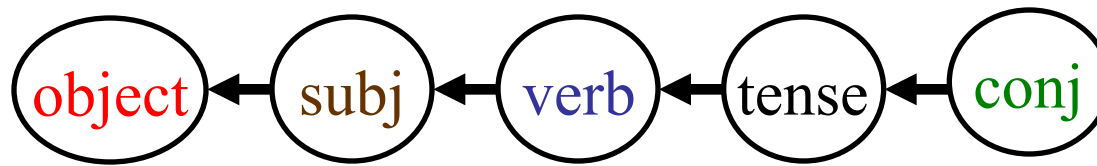
and+for+the+library+plural

And for the libraries

- Morphotactics (e.g. ل+ال → لل)
- Arabic *Broken Plurals* (templatic)

Inflectional Morphology

Verbs



فقلناها

/faqulnāhā/

ف + قال + نا + ها

fa+qul+na+hā

so+said+we+it

So we said it.

وسنقولها

/wasanaqūluhā/

و + سن + قول + ها

wa+sa+na+qūl+u+hā

and+will+we+say+it

And we will say it

- Morphotactics
- Subject conjugation (suffix or circumfix)

Morphological Ambiguity

- **Derivational ambiguity**
basis/principle/rule, military base, Qa'ida/Qaeda/Qaida

قاعدة

- **Inflectional ambiguity**
- You write/she writes

تكتب

- **Segmentation ambiguity**
and+grandfather : وجد ; he found : وجد •

- **Spelling ambiguity**
- Optional diacritics
kātib/ writer , /kātab/ to correspond/ : كاتب •

- Suboptimal spelling

- Hamza dropping: ا → اُ , اِ

- Undotted ta-marbuta: ه → ة

- Undotted final ya: ي → ى

Morphology Summary

- Rich complex morphology
 - Templatic, concatenative, derivational, inflectional
 - wbHsnAthm
 - w+b+Hsn+At+hm
 - and by virtue(s) their
 - Verbs are marked for tense, person, gender, aspect, mood, voice
 - Nominals are marked for case, number, gender, definiteness
- Orthography is underspecified for short vowels and consonant doubling (diacritics)

Syntax

- Pro-drop language
 - Akl AlbrtqAl '[he] ate the orange(s)'
 - *hw* Akl AlbrtqAl 'he ate the orange(s)'
- Relative free word order
 - VSO, SVO, OVS, etc.
 - The canonical order is VSO, dialects are more SVO
 - In Arabic Treebank v3.2 we observe equal distribution of SVO (35%) and VSO (35%) and pro-drop (30%)
- Complex noun phrases expressing possession
'idafa constructions
 - mlk AlArdn 'king_INDEF Jordan'
king of Jordan

Characteristics relevant for SRL

- Typical underspecification of short vowels masks morphological features such as case and agreement
 - Example:

rjl Albyt Alkbyr

Man__{masc} the-house__{masc} the-big__{masc}

“the big man of the house” or “the man of the big house”

Characteristics relevant for SRL

- Typical underspecification of short vowels masks morphological features such as case and agreement
 - Example:

rjlu Albyti Alkbyri

Man__{masc-Nom} the-house__{masc-Gen} the-big__{masc-Gen}

the man of the big house

Characteristics relevant for SRL

- Typical underspecification of short vowels masks morphological features such as case and agreement
 - Example:

rjl Albyti Alkbyru

Man_masc-Nom the-house_masc-Gen the-big_masc-Nom

the big man of the house

Characteristics relevant for SRL

- Idafa constructions make indefinite nominals syntactically definite hence allowing for agreement, therefore better scoping
 - Example:

[rjlu Albyti] Alkbyru

Man_{_masc-Nom-Def} the-house_{_masc-Gen} the-big_{_masc-Nom-Def}

the big man of the house

Characteristics relevant for SRL

- Passive constructions are hard to detect due to underspecified short vowels marking passivization inflection.
- Best automatic systems are at 68% acc.

- Example:

qtl **Emr** bslAH qAtl....

[He]_{pro-drop} **killed** **Amr** by a deadly weapon...

Amr **killed** by a deadly weapon ...

Amr **was killed** by a deadly weapon

Characteristics relevant for SRL

- Passive constructions are hard to detect due to underspecified short vowels marking passivization inflection.
- Hence

- Example:

qatal Emra_{_ACC_ARG1} bslAHiK qAtliK....

[He]_{pro-drop} killed Amr_{_ACC_ARG1} by a deadly weapon...

Amr killed by a deadly weapon ...

Amr was killed by a deadly weapon

Characteristics relevant for SRL

- Passive constructions are hard to detect due to underspecified short vowels marking passivization inflection.
- Hence

- Example:

qatal Emru__{NOM_ARGO} bslAHiK qAtliK....

[He]_{pro-drop} killed Amr by a deadly weapon...

Amr__{NOM_ARGO} killed by a deadly weapon ...

Amr was killed by a deadly weapon

Characteristics relevant for SRL

- Passive constructions are hard to detect due to underspecified short vowels marking passivization inflection.
- Hence

- Example:

qutil Emru _{_NOM_ARG1} bslAHiK qAtliK....

[He]_{pro-drop} killed Amr by a deadly weapon...

Amr killed by a deadly weapon ...

Amr _{_NOM_ARG1} was killed by a deadly weapon

Characteristics relevant for SRL

- Passive constructions differ from English in that they can not have an explicit non-instrument underlying subject, hence only ARG1 and ARG2. ARG0 are not allowed.

- Example:

qutil Emru bslAHiK qAtliK

*qutil [Emru]_{ARG1} [bslmY]_{ARG0}

*[Amr]_{ARG1} was killed [by Salma]_{ARG0}

Characteristics relevant for SRL

- Passive constructions differ from English in that they can not have an explicit non-instrument underlying subject, hence only ARG1 and ARG2. ARG0 are not allowed.

- Example:

qutil [Emru]_{ARG1} [bslAHiK qAtliK]_{ARG2}

[Amr]_{ARG1} was killed [by a deadly weapon]_{ARG2}

Characteristics relevant for SRL

- Relative free word order combined by agreement patterns between Subject and Verb could be helpful when explicit yet confusing with absence of case and passive marker and pro-drop
- VSO = Gender agreement only between V and S
- SVO = Gender and Number agreement

Our Approach

ACL 2008

In collaboration with

Alessandro Moschitti, Daniele Pighin

Supervised SRL

- We need training data
 - Data annotated with propositions
- Hence the need for an Arabic Propbank

What is a Propbank?

- A proposition bank annotates propositions identifying predicates and their arguments and associating them with their relevant semantic roles
- *Example*
 - Lexical Semantics: [John]_{Agent} loved [Mary]_{Theme}
 - Framenet: [John]_{Lover} loved [Mary]_{Lovee}
 - PropBank: [John]_{ARG0} loved [Mary]_{ARG1}
- *Crucially roles do not vary with surface syntax,*
 - [Mary]_{ARG1} was loved by [John]_{ARG0}

Semantic Role Labeling Steps

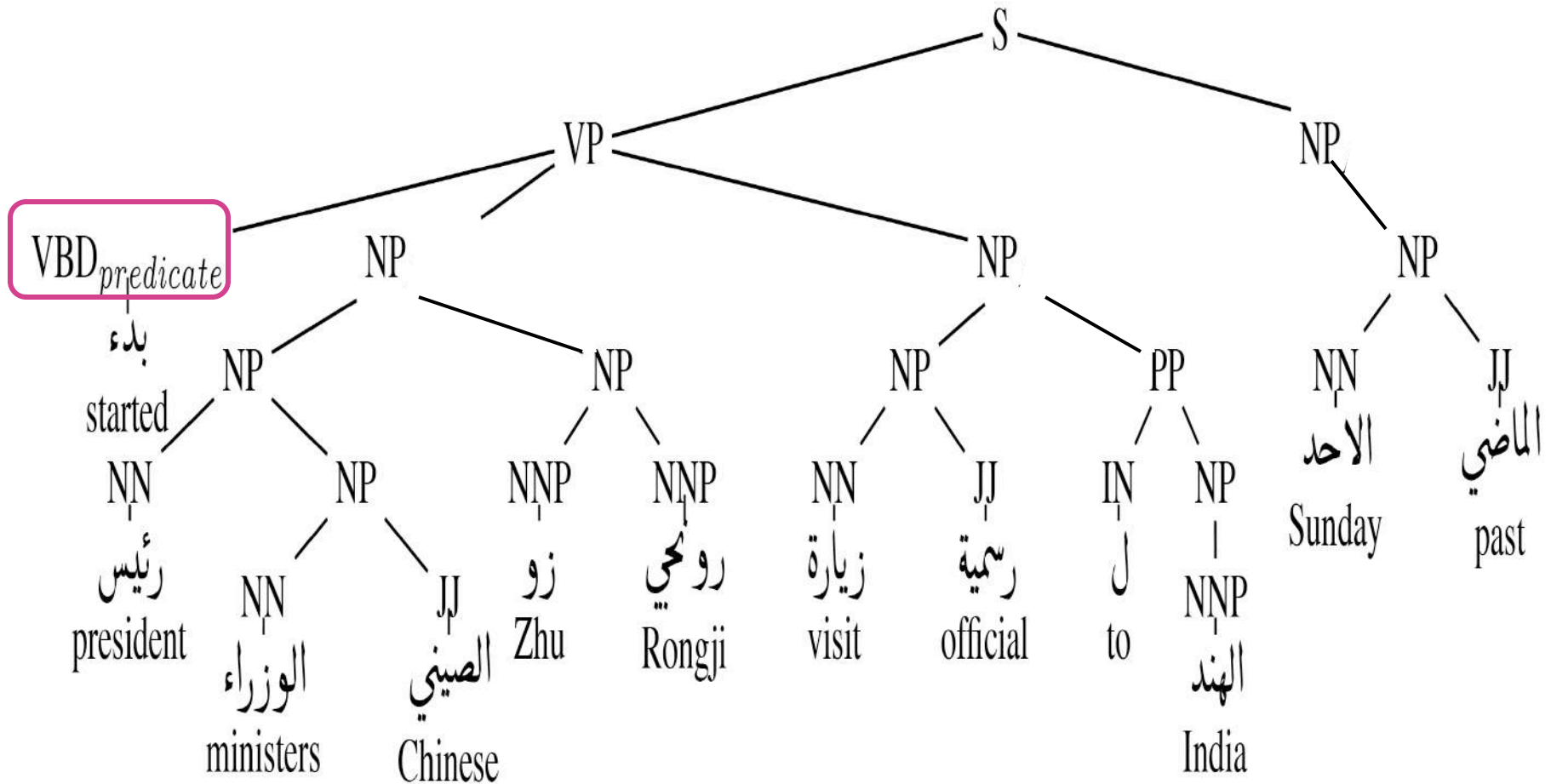
- Given a sentence and an associated syntactic parse
- An SRL system identifies the arguments for a given predicate
- The arguments are identified in two steps
 - Argument boundary detection
 - Argument role classification
- For the overall system we apply a heuristic for argument label conflict resolution
 - one label per argument

The Sentence

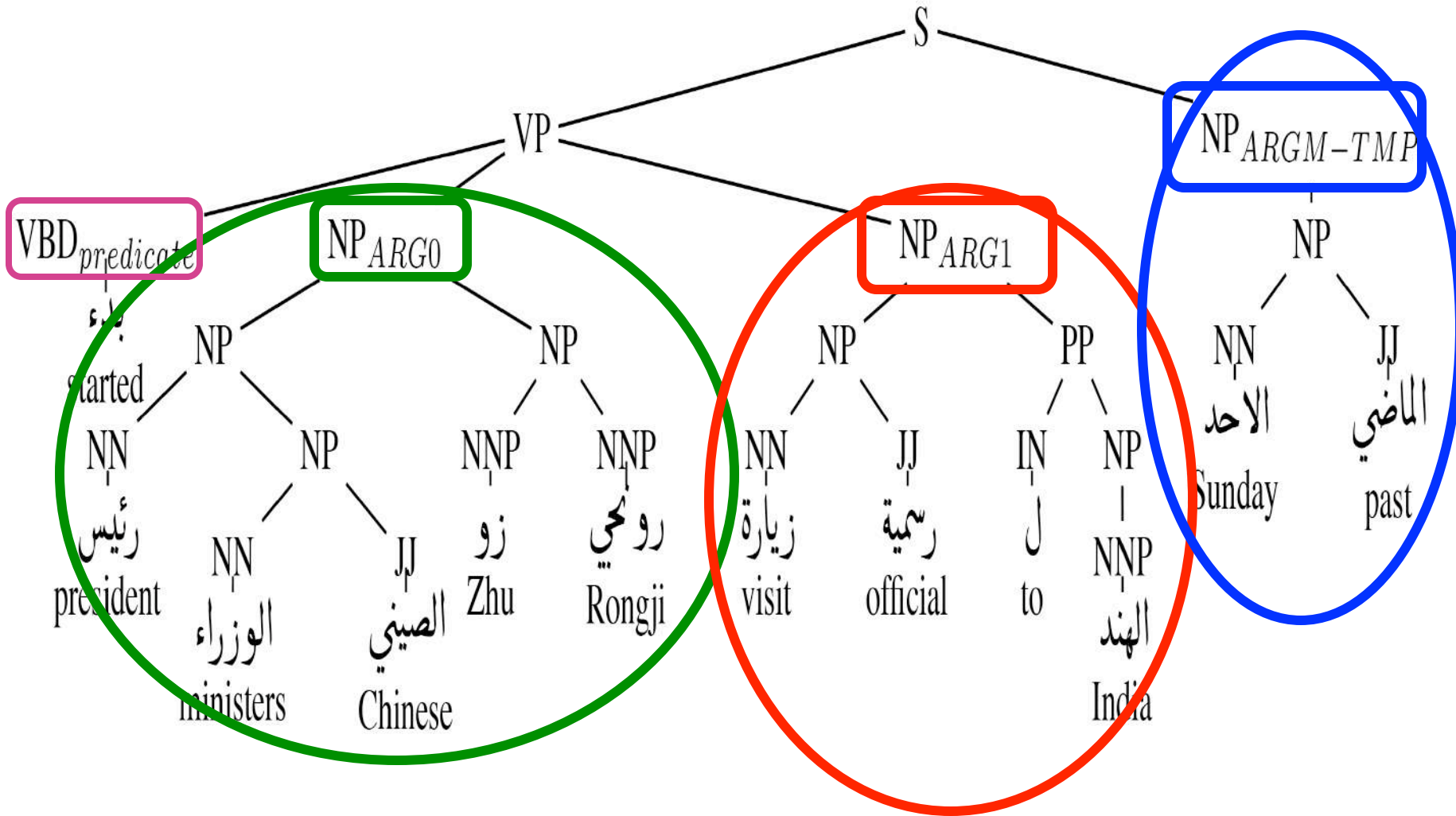
بدأ رئيس الوزراء الصيني زو رونغجي زيارة رسمية لهند الأحد الماضي

The Chinese Prime Minister Zho Rongji started an official visit to India last sunday

The Parse Tree



Role Classification



Our Approach

- Experiment with different kernels
- Experiment with Standard Features (similar to English) and rich morphological features specific to Arabic

Different Kernels

- Polynomial Kernels (1-6) with standard features

- Tree Kernels

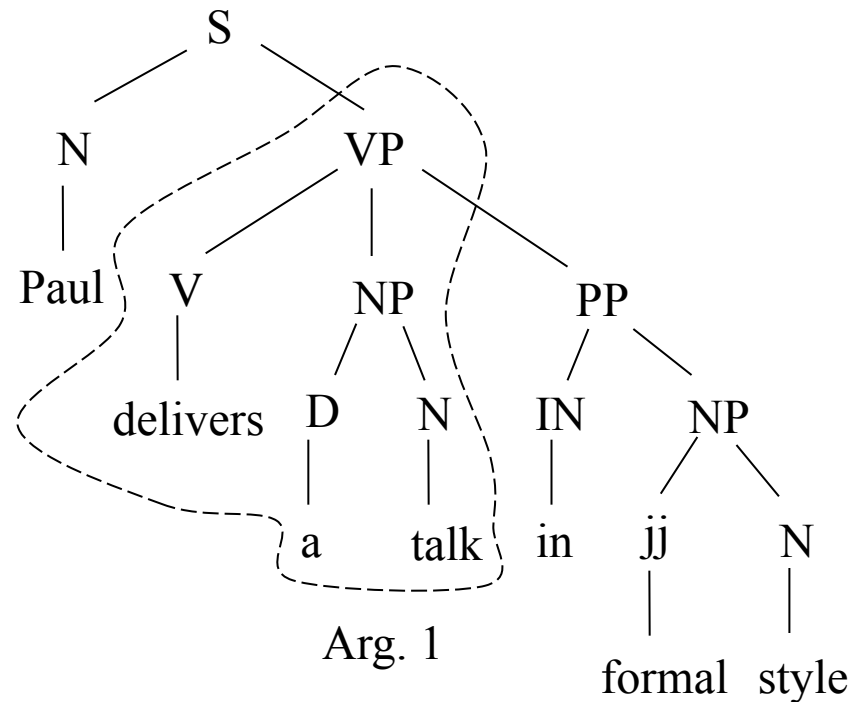
$$K_T(t_1, t_2) = \sum_{n_1 \in N_{t_1}} \sum_{n_2 \in N_{t_2}} \Delta(n_1, n_2)$$

Where N_{t_1} and N_{t_2} are the sets of nodes in t_1 and t_2 , and $\Delta(.)$ evaluates the common substructures rooted in n_1 and n_2

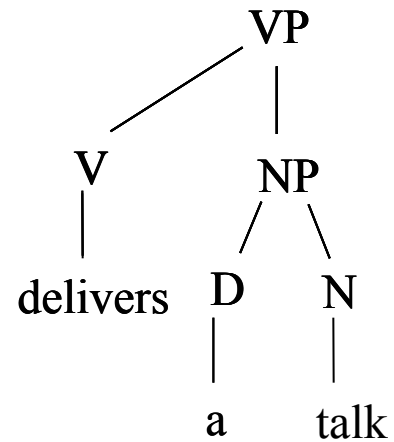
$$\Delta(n_1, n_2) = \sum_{i=1}^{|\bar{\mathcal{F}}|} I_i(n_1) I_i(n_2)$$

Argument Structure Trees (AST)

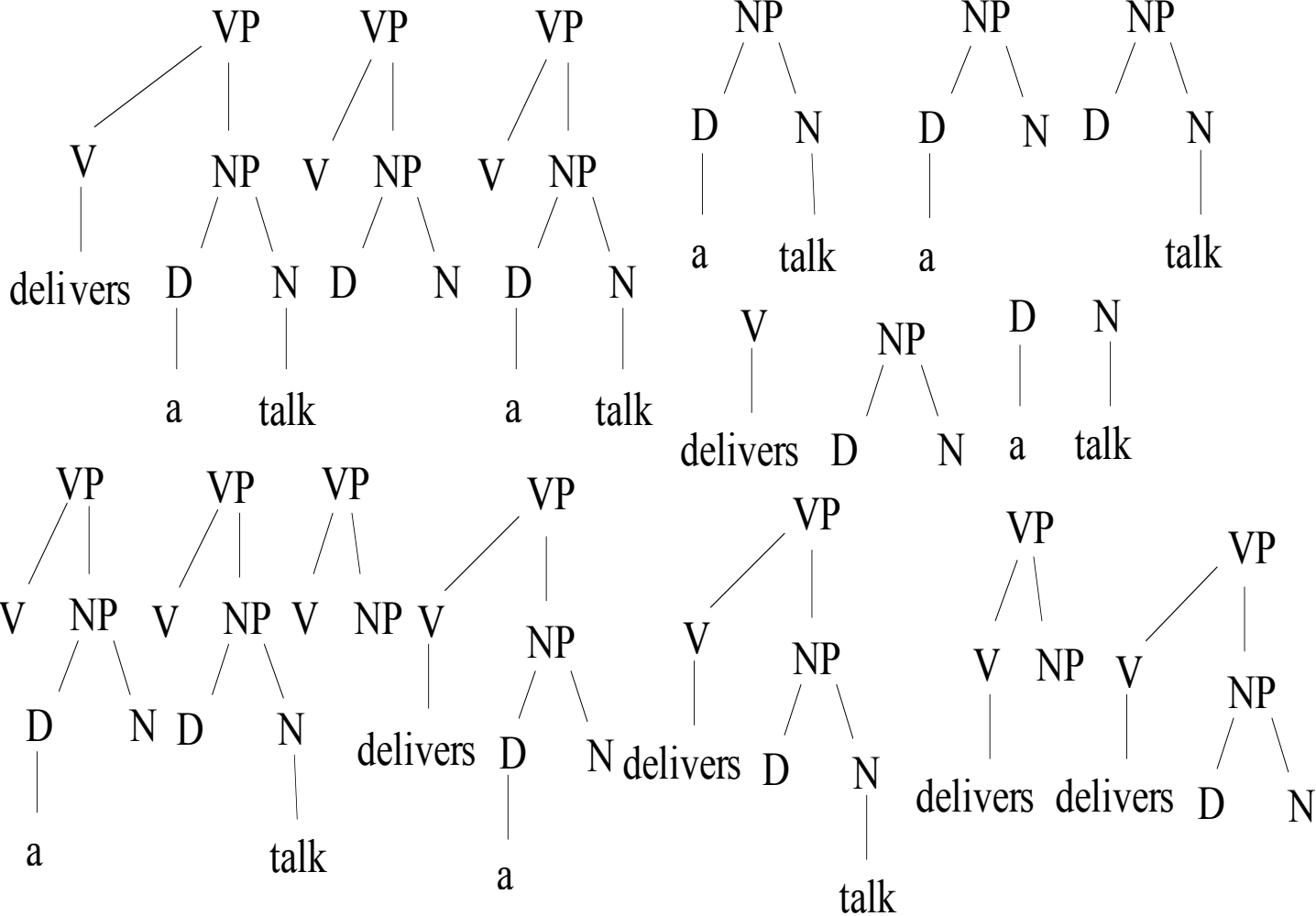
Defined as the minimal subtree encompassing the predicate and one of its arguments



Tree Substructure Representations

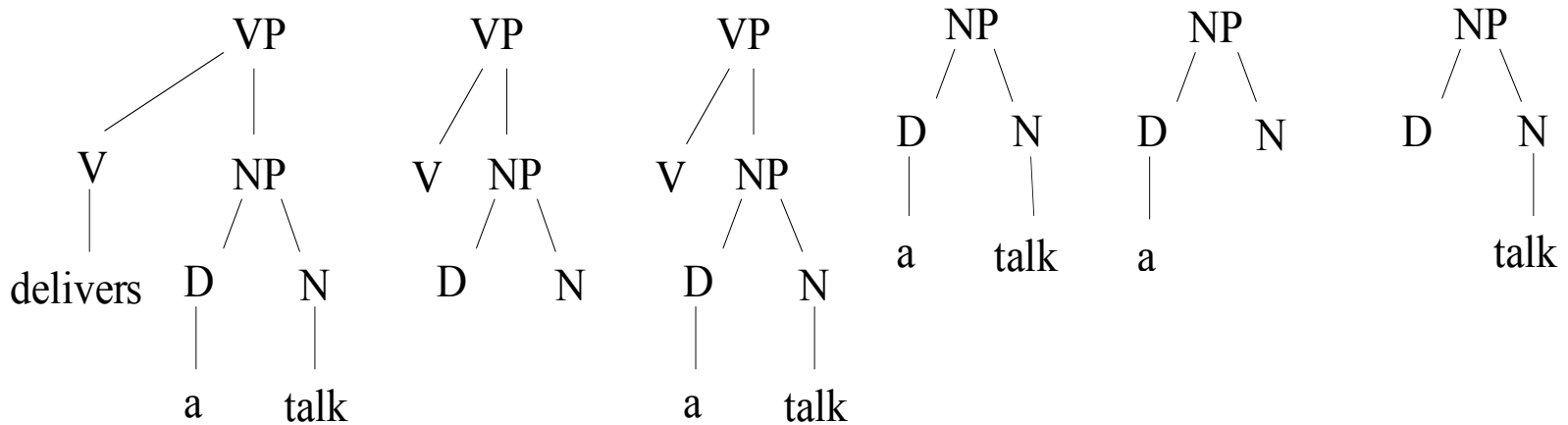


The overall set of AST substructures



Explicit feature space

$$\vec{x} = (0, \dots, 1, \dots, 0, \dots, 1, \dots, 0, \dots, 1, \dots, 0, \dots, 1, \dots, 0, \dots, 1, \dots, 0, \dots, 1, \dots, 0)$$



- $\vec{x} \cdot \vec{z}$ counts the number of common substructures

Standard Features

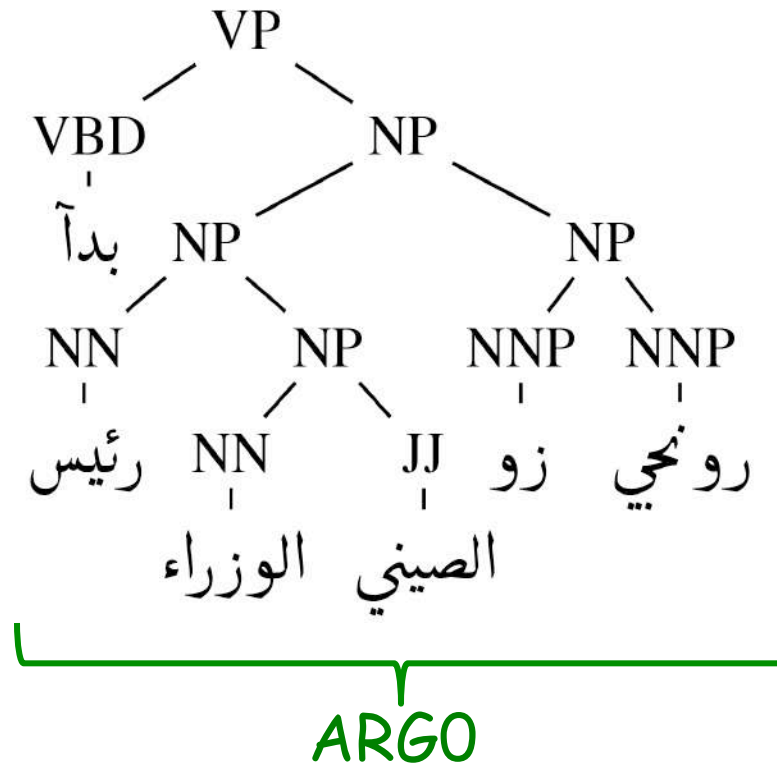
- Predicate: Lemmatization of the predicate
- Path: Syntactic path linking the predicate and an argument NN↑NP↑VP↓VBD
- Partial Path: Path feature limited to the branching of arg
- No Direction path without the traversals
- Phrase type
- Last and first POS of words in the arguments
- Verb subcategorization frame: production expanding the predicate parent node
- Position of the argument relative to predicate
- Syntactic Frame: positions of the surrounding NPs relative to predicate

Extended Features for Arabic

Definiteness, Number, Gender, Case, Mood,
Person, Lemma (vocalized), English Gloss,
Unvocalized surface form, Vocalized
Surface form

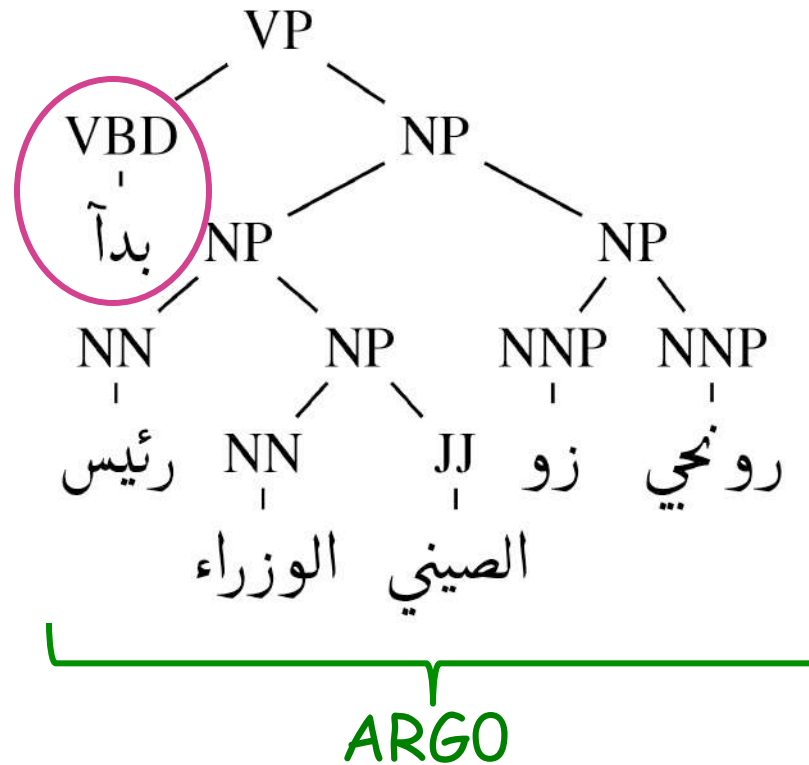
- Expanded the leaf nodes in AST with 10 attribute value pairs creating EAST

Arabic AST



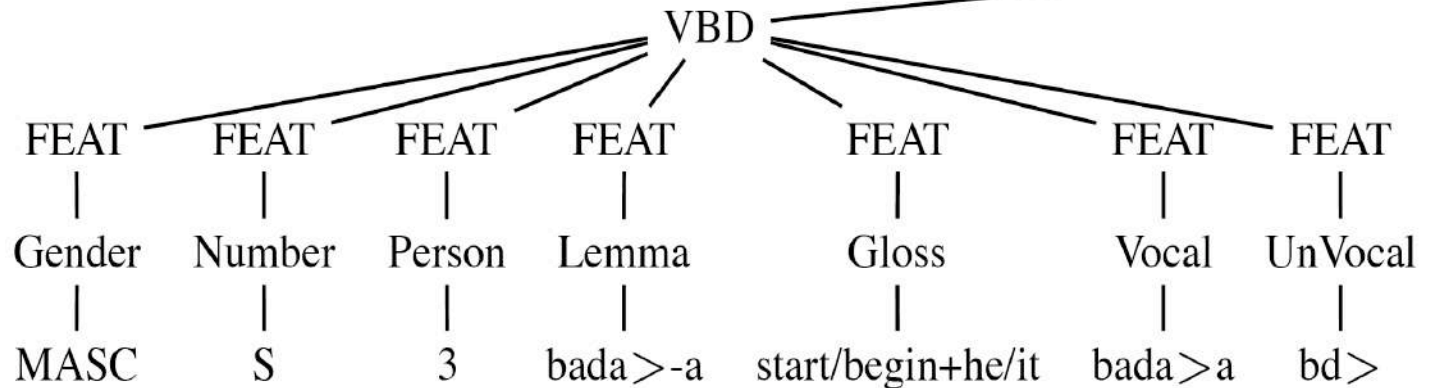
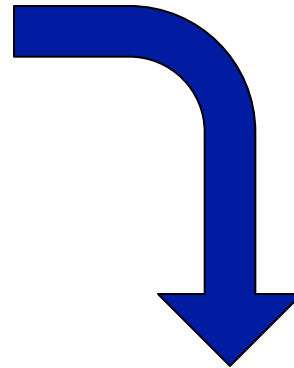
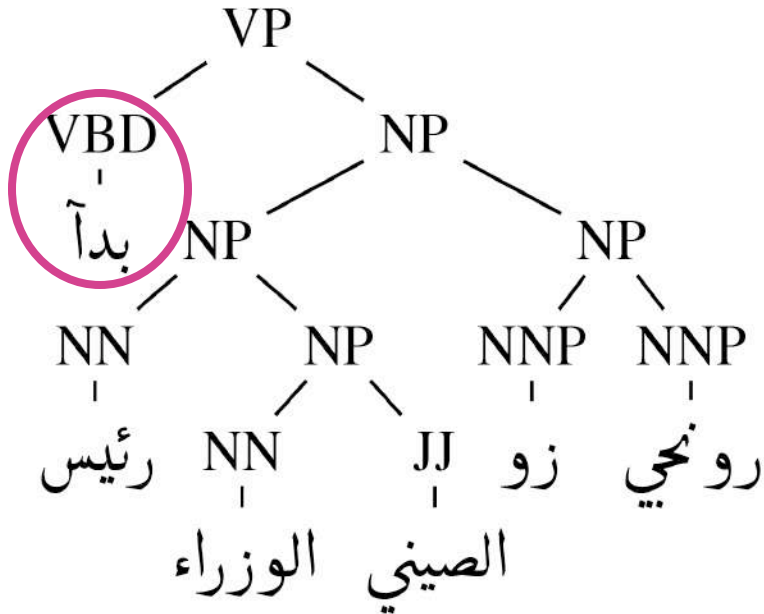
Sample AST from our example

Arabic AST



Sample AST from our example

Extended AST (EAST)



Experiments & Results

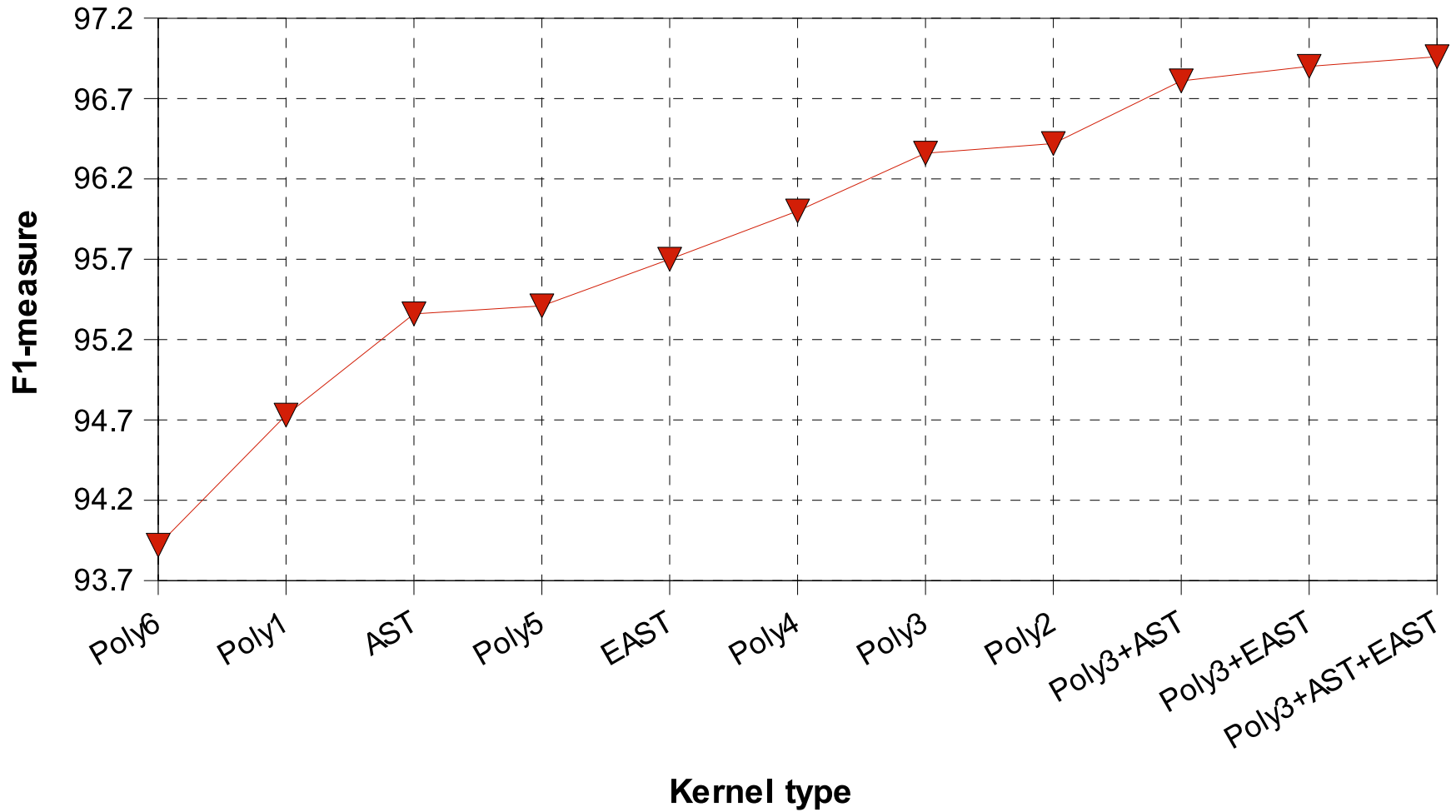
Experimental Set Up

- SemEval 2007 Task 18 data set, Pilot Arabic Propbank
- 95 most frequent verbs in ATB3v2
- Gold parses, Unvowelized, Bies reduced POS tag set (25 tags)
- Num Sentences: Dev (886), Test (902), Train (8402)
- 26 role types (5 numbered ARGs)

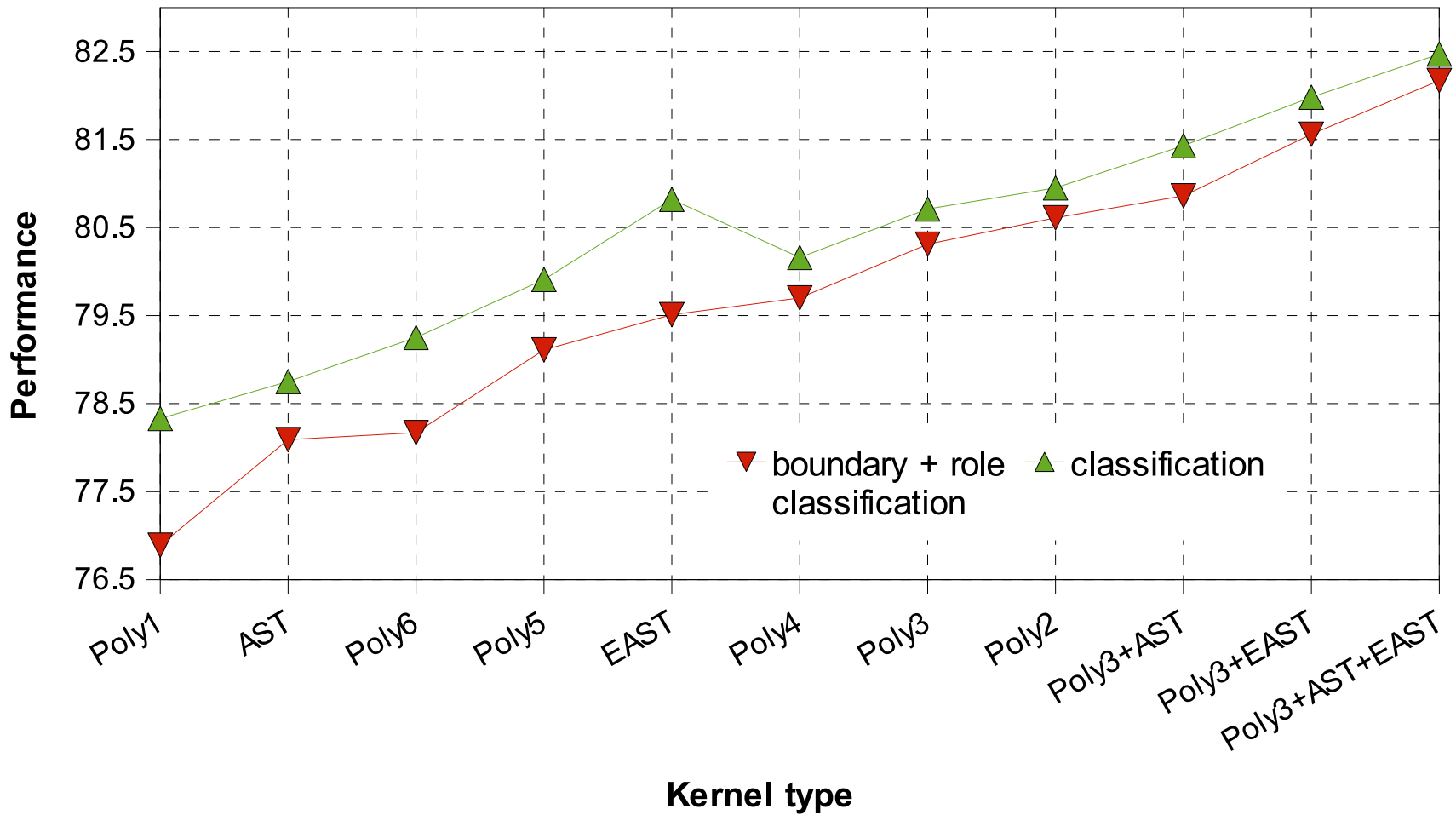
Experimental Set Up

- Experimented only with 350k examples
- We use the SVM-Light TK Toolkit (Moschitti, 2004, 2006) with SVM light default parameters
- Evaluation metrics of precision, recall and F measure are obtained using the CoNLL evaluator

Boundary Detection Results



Role Classification Results



Overall Results

	P3	AST	EAST	AST+ P3	EAST+ P3	AST+ EAST+ P3
P	81.73	80.33	81.7	81.73	82.46	83.08
R	78.93	75.98	77.42	80.01	80.67	81.28
F ₁	80.31	78.09	79.51	80.86	81.56	82.17

Observations-BD

- AST and EAST don't differ much for boundary detection
- AST+EAST+ Poly (3) gives best BD results
- AST and EAST perform significantly better than Poly (1)

Observations - RC & SRL

- For classification, EAST is 2 absolute f-score points better than AST
- AST is better than Poly(1) and EAST is better than Poly(1) and AST for both classification and overall system
- Poly 2 and 3 are similar to EAST in classification
- AST+EAST+best Poly, Poly(3), yields best classification results
- Best results yielded are for ARG0 and ARG1
- ARG1 because of passive cases in Arabic is harder than in English

More observations

- Explicitly encoding the rich morphological features helps with SRL in Arabic
- Tree Kernels is indeed a feasible way of dealing with large feature spaces that are structural in nature
- Combining kernels yields better results

Current Directions

- Experiment with richer POS tag sets
- Experiment with automatic parses
- Experiment with different syntactic formalisms
- Integrate polynomial kernels with tree kernels
- Experiment with better conflict resolution approaches

Task of NER

What is NER

A secular pilgrimage:

Since fourteen years Amzeel visits Tangier every year and she got herself a house there, also Amzeel explains her attraction saying "no wonder, since visiting Tangier to me is a secular pilgrimage which happens every year and lasts for long", she added "every single thing small or big in Tangier and Morocco is very beautiful, takes my breath away and makes me forget my home country and makes me wish I could stay here forever".



Ph.D. Abdelnabi Serokh a professor in Abdelmalek Essaadi University in Tangier, who keeps company to the artist, will give an analytic study of her works which is subject of the book they are writing together.

Input
←

Output
→

A secular pilgrimage:

Since fourteen years Amzeel visits Tangier every year and she got herself a house there, also Amzeel explains her attraction saying "no wonder, since visiting Tangier to me is a secular pilgrimage which happens every year and lasts for long", she added "every single thing small or big in Tangier and Morocco is very beautiful, takes my breath away and makes me forget my home country and makes me wish I could stay here forever".



Ph.D. Abdelnabi Serokh a professor in Abdelmalek Essaadi University in Tangier, who keeps company to the artist, will give an analytic study of her works which is subject of the book they are writing together.

What is NER

رحلة حج علمانية

ومنذ أربع عشرة سنة وأمزيل تزور طنجة كل سنة حتى حصلت على بيت لها فيها، وتفسر أمزيل هذا الانجذاب قائلة "لا عجب، فزيارة طنجة بالنسبة لي رحلة حج علمانية تتكرر كل صيف وتنوم طويلا"، مضيئة "كل شيء صغير وكبير في طنجة والمغرب جميل جدا يأخذ بلبي وأنسى بلدي متمنية أن أبقى طول عمري هنا".


و يرافق الفنانة الدكتور عبد النبي صروخ الأستاذ بجامعة عبد الملك السعدي بطنجة، الذي سيقدم قراءة تحليلية لأعمالها ضمن كتاب مشترك معها.



Input



Output



رحلة حج علمانية

ومنذ أربع عشرة سنة وأمزيل تزور طنجة كل سنة حتى حصلت على بيت لها فيها، وتفسر أمزيل هذا الانجذاب قائلة "لا عجب، فزيارة طنجة بالنسبة لي رحلة حج علمانية تتكرر كل صيف وتنوم طويلا"، مضيئة "كل شيء صغير وكبير في طنجة والمغرب جميل جدا يأخذ بلبي وأنسى بلدي متمنية أن أبقى طول عمري هنا".

و يرافق الفنانة الدكتور عبد النبي صروخ الأستاذ بجامعة عبد الملك السعدي بطنجة، الذي سيقدم قراءة تحليلية لأعمالها ضمن كتاب مشترك معها.



NER as a Classification Task



رحلة حج علمانية
ومنذ أربع عشرة سنة وأمزيل تزور طنجة كل سنة حتى حصلت على بيت لها فيها، وتفسر أمزيل هذا الانجذاب قائلة "لا عجب، فزيارة طنجة بالنسبة لي رحلة حج علمانية تتكرر كل صيف وتدوم طويلا"، مضيفة "كل شيء صغير وكبير في طنجة والمغرب جميل جدا يأخذ بلبي وأنسى بلدي مثمانية أن أبقى طول عمري هنا".
ويرافق الفنانة الدكتور عبد النبي صبيح الأستاذ بجامعة عبد المالك السعدي بطنجة الذي سيقدم قراءة تحليلية لاصحائها ضمن كتاب مشترك معها.

University	B-ORG	جامعة
Abd	I-ORG	عبد
Almalek	I-ORG	المالك
Esaâdi	I-ORG	السعدي
In	O	ب
Tangier	B-LOC	طنجة
,	O	,
who	O	الذي
...

Peculiarities and Challenges for Arabic NER

Three main issues

A secular pilgrimage:

Since fourteen years Amzeel visits Tangier every year and she got herself a house there, also Amzeel explains her attraction saying "no wonder, since visiting Tangier to me is a secular pilgrimage which happens every year and lasts for long", she added "every single thing small or big in Tangier and Morocco is very beautiful, takes my breath away and makes me forget my home country and makes me wish I could stay here forever".



Ph.D. Abdelnabi Serokh a professor in Abdelmalek Essaadi University in Tangier, who keeps company to the artist, will give an analytic study of her works which is subject of the book they are writing together.



رحلة حج علمانية

ومنذ أربع عشرة سنة وأمزيل تزور طنجة كل سنة حتى حصلت على بيت لها فيها، وتفسر أمزيل هذا الانجذاب قائلة "لا عجب، فزيارة طنجة بالنسبة لي رحلة حج علمانية تتكرر كل صيف وتدوم طويلا"، مضيفة "كل شيء صغير وكبير في طنجة والمغرب جميل جدا يأخذ بلبي وأنسى بلدي متمنية أن أبقى طول عمري هنا".

وبرافق الفنانة الدكتور عبد النبي صروح

الأستاذ بجامعة عبد المالك السعدي بطنجة،

الذي سيقدم قراءة تحليلية لأعمالها ضمن كتلب مشترك معها.

The Ph.D. Abdelnabi Serokh a professor in
Abdelmalek Essaâdi University in Tangier

الدكتور عبد النبي صروخ الأستاذ بجامعة
عبد المالك السعدي بطنجة

Lack of Short Vowels

Th Ph.D. Abdlnbi Srkh a prfssr n AbdImalek
Essâdi Unvrsty n Tangr

الدكتور عبد النبي صروخ الأستاذ بجامعة
عبد المالك السعدي بطنجة

Increases ambiguity

No Capitalization of Arabic Letters

th ph.d. abdlnbi srkh a prfssr n abdlmalek
essâdi unvrsty n tangr

الدكتور عبد النبي صروخ الأستاذ بجامعة
عبد المالك السعدي بطنجة

NE detection becomes harder

Complex/Rich Morphology

thph.d. abdlnbi srkh aprfssr nabdlmalek
essâdi unvrsty ntangr

الدكتور عبد النبي صروخ الأستاذ بجامعة
عبد المالك السعدي بطنجة

Increases data sparseness

Supervised ML for NER

- Use a wide range of features: contextual, lexical, gazetteers, syntactical and morphological
- Use ME, CRFs and SVMs
- Use a wrapper incremental feature selection approach in order to optimize the feature-set
- Evaluate the approaches on many data-sets

Features

- Context: -/+n lexical context and -n tag context
- Lexical: $C_1 C_2 C_3 \dots C_{n-2} C_{n-1} C_n$

$LEX1=C_1; LEX2=C_1C_2; LEX3=C_1C_2C_3;$
 $LEX4=C_n; LEX5=C_{n-1}C_n; LEX6=C_{n-2}C_{n-1}C_n$

- Syntactical: POS-tag and BPC
- Nationality
- Capitalization of Corresponding English Translation
- Morphological: a tool for Morphological Analysis and Disambiguation for Arabic (MADA)

Feature Ranking

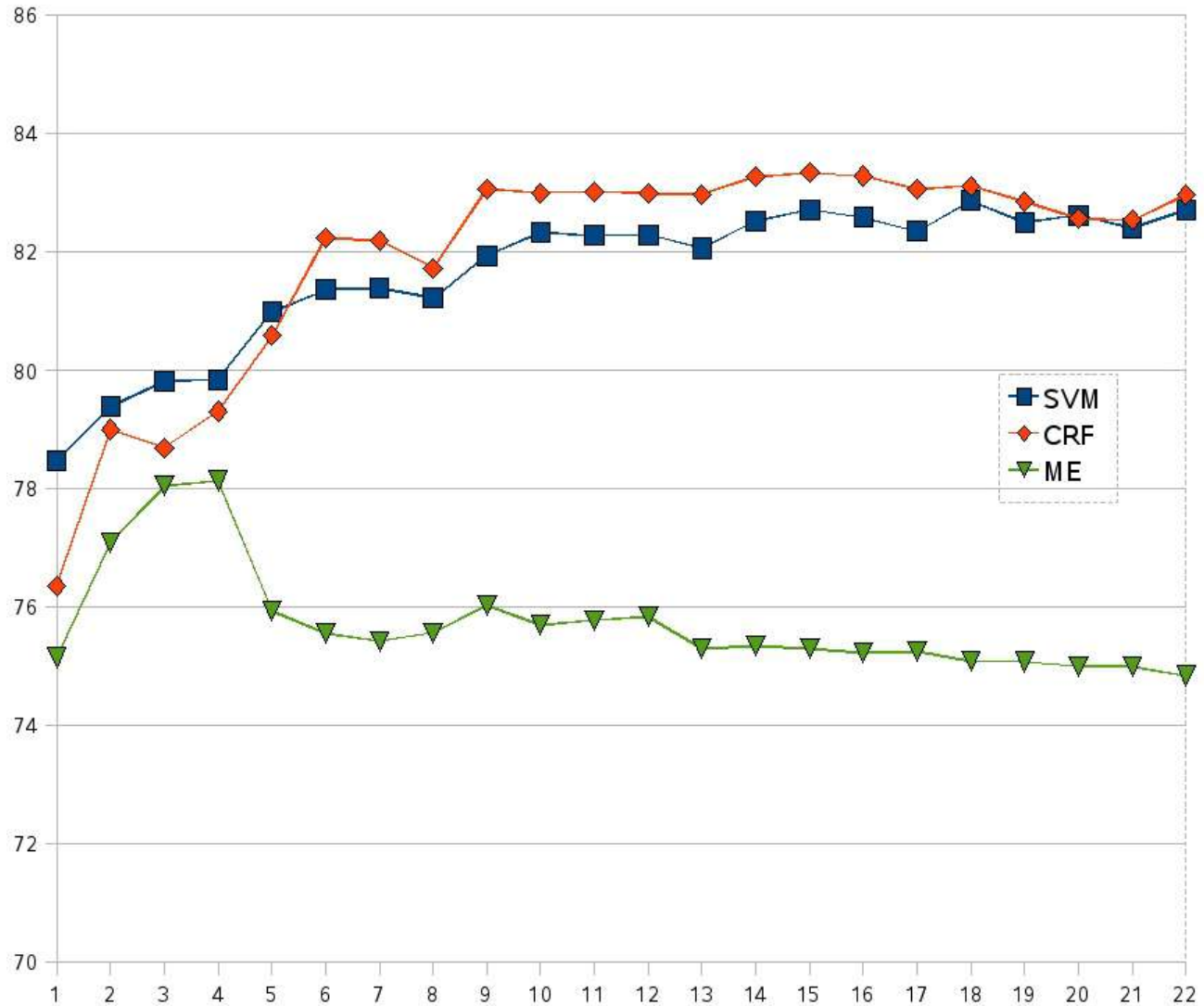
M1= article
M2= aspect
M3= grammatical case
M4= clitic
M5= conjunction
M6= definiteness
M7= mood
M8= number
M9= particle
M10= person
M11= voice

Rank	Feature	Rank	Feature
1	POS	12	NAT
2	CAP	13	<i>LEX</i> ₁
3	<i>M</i> ₂	14	<i>LEX</i> ₄
4	<i>M</i> ₉	15	<i>M</i> ₃
5	<i>LEX</i> ₆	16	<i>M</i> ₈
6	<i>LEX</i> ₃	17	<i>M</i> ₆
7	<i>M</i> ₄	18	<i>LEX</i> ₂
8	BPC	19	<i>LEX</i> ₅
9	GAZ	20	<i>M</i> ₅
10	<i>M</i> ₁	21	<i>M</i> ₇
11	<i>M</i> ₁₁	22	<i>M</i> ₁₀

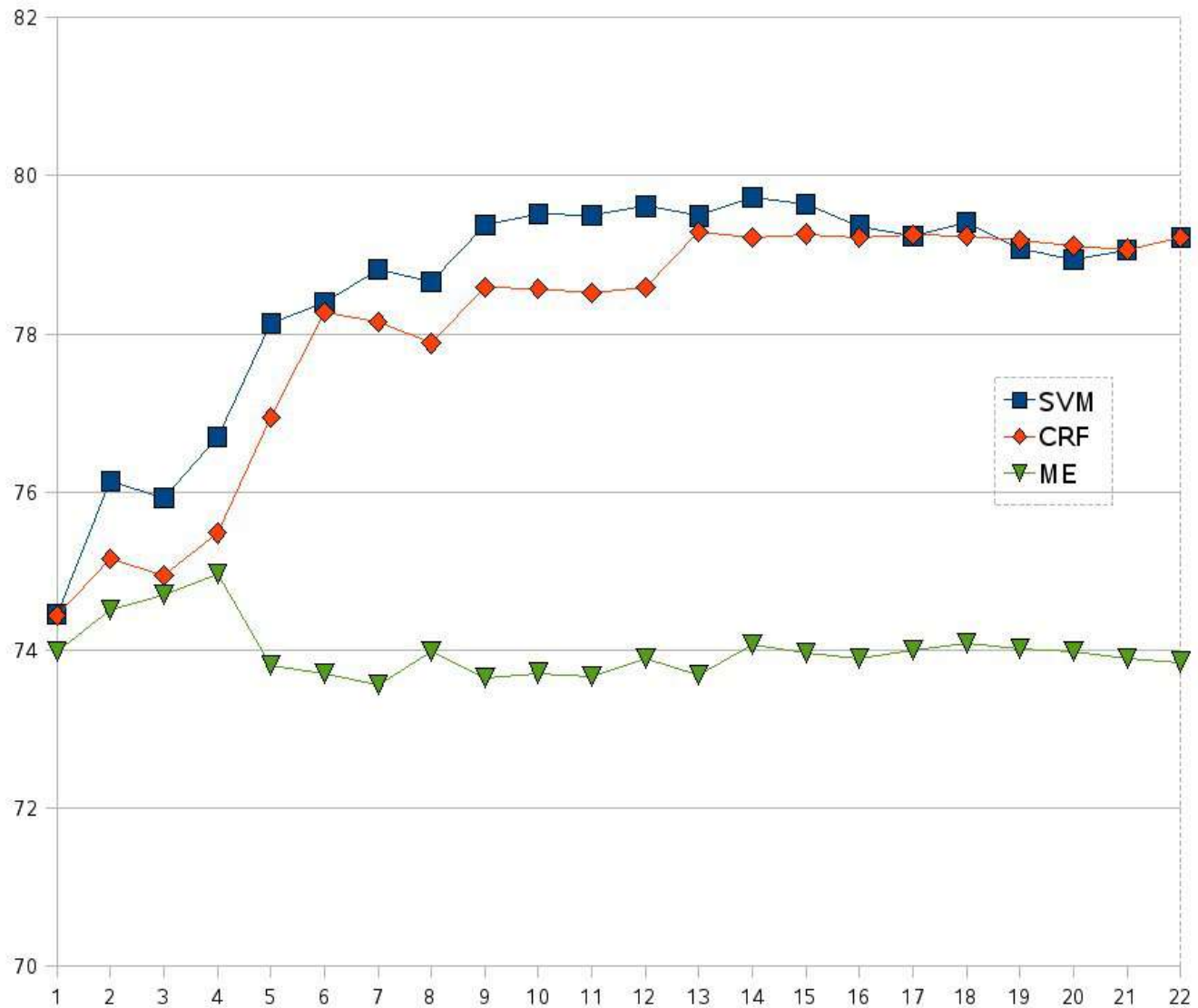
Parameter Setting Experiments

	-1/+1	-2/+2	-3/+3	-4/+4
CXT+UNTOK	71.66	67.45	61.73	57.49
CXT+TOK	74.86	72.24	67.71	64

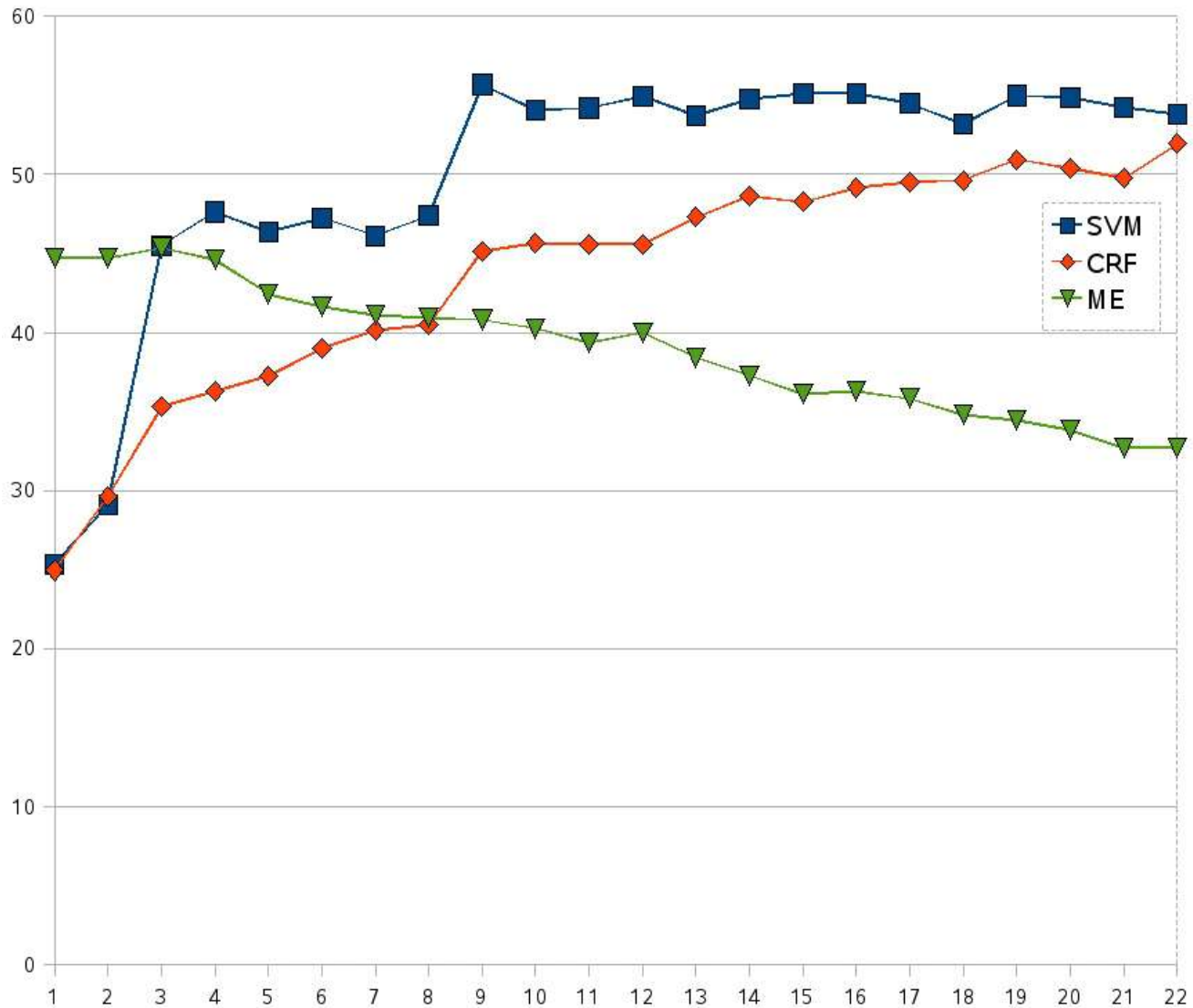
Best Results



2003 NewsWire



2005 WebLogs



Results

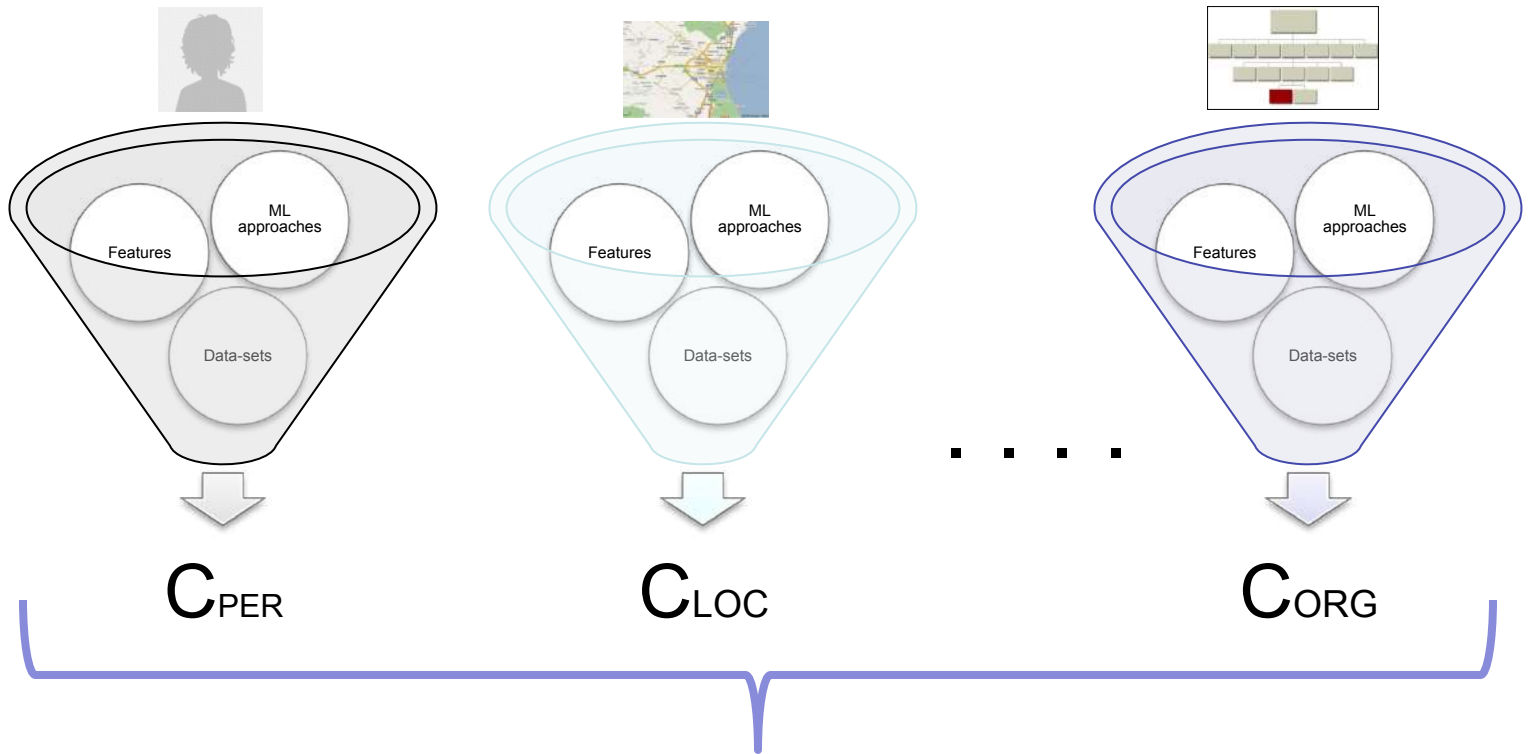
Corpus	genre	Baseline	Best						All Features		
			SVMs		ME		CRFs		SVMs	ME	CRFs
			<i>N</i>	<i>F</i>	<i>N</i>	<i>F</i>	<i>N</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>
ANERcorp 2.0	NW	31.5	14	81.04	3	77.9	12	80.36	80.4	76.8	79.8
ACE 2003	BN	74.78	15	82.72	3	78.05	15	83.34	82.71	74.84	82.94
	NW	69.08	14	79.72	3	74.56	13	79.52	79.21	73.84	79.11
ACE 2004	BN	62.02	16	77.61	2	73.34	13	77.03	76.43	69.44	76.96
	NW	52.23	14	74.13	3	68.13	12	74.53	73.4	63.13	73.47
	ATB	64.23	15	75.43	2	69.95	13	75.51	75.34	64.66	75.48
ACE 2005	BN	71.06	15	82.02	3	77.67	14	81.87	81.47	75.71	81.1
	NW	58.63	15	76.97	3	70.31	13	77.06	76.19	67.41	75.67
	WL	27.66	12	55.69	2	44.96	14	53.91	53.81	32.66	51.81

Results Discussion & Error Analysis

- Per class results:

Class	<i>SVMs</i>		<i>ME</i>		<i>CRFs</i>	
	<i>Best</i>	<i>All</i>	<i>Best</i>	<i>All</i>	<i>Best</i>	<i>All</i>
<i>FAC</i>	13.33	13.33	23.64	24	13.34	0
<i>LOC</i>	86.66	87.04	83.32	81.29	87.27	87.03
<i>ORG</i>	54.36	51.31	47.56	49.53	51.35	49.12
<i>PER</i>	81.55	81.43	76.16	67.61	82.70	82.83
<i>Overall</i>	82.72	82.71	78.05	74.84	83.34	82.94

Combining Classifiers



Outcomes Combination

Feature Selection Approach

Split data into train, dev and test

```
graph TD; A[Split data into train, dev and test] --> B[Measure indiv. Impact of each feat. for each ML approach and NE class]; B --> C[Rank the features for each NE class using Fuzzy Borda Voting Scheme (FBVS)]; C --> D[Select a feature-set and a ML approach for each NE class];
```

Measure indiv. Impact of each feat. for each ML approach and NE class

Rank the features for each NE class using **Fuzzy Borda Voting Scheme (FBVS)**

Select a feature-set and a ML approach for each NE class

Results

		<i>ACE 2003</i>		<i>ACE 2004</i>			<i>ACE 2005</i>		
		<i>BN</i>	<i>NW</i>	<i>BN</i>	<i>NW</i>	<i>ATB</i>	<i>BN</i>	<i>NW</i>	<i>WL</i>
	<i>FreqBaseline</i>	73.74	67.61	62.17	51.67	62.94	70.18	57.17	27.66
	<i>MLBaseline_{SVMs}</i>	80.58	76.37	74.21	71.11	73.14	79.3	73.9	54.68
	<i>MLBaseline_{CRFs}</i>	81.02	76.18	74.67	71.8	73.04	80.13	74.75	55.32
dev	<i>Best Feat-set/ML</i>	83.41	79.11	76.9	72.9	74.82	81.42	76.07	54.49
	<i>All Feats. SVMs</i>	81.79	77.99	75.49	71.8	73.71	80.87	75.69	53.73
	<i>All Feats. CRFs</i>	81.76	76.6	76.26	71.85	74.19	79.66	74.83	36.11
test	<i>Best Feat-set/ML</i>	83.5	78.9	76.7	72.4	73.5	81.31	75.3	57.3
	<i>All Feats. SVMs</i>	81.76	77.27	69.96	71.16	59.23	81.1	72.41	55.58
	<i>All Feats. CRFs</i>	81.37	75.89	75.73	72.36	74.21	80.16	74.43	27.36

	<i>BN</i>		<i>NW</i>		<i>ATB</i>		<i>WL</i>	
	<i>N</i>	<i>ML</i>	<i>N</i>	<i>ML</i>	<i>N</i>	<i>ML</i>	<i>N</i>	<i>ML</i>
Person	12	SVM	14	SVM	9	SVM	11	SVM
Location	10	SVM	7	SVM	16	CRF	14	SVM
Organization	9	CRF	6	CRF	10	CRF	12	CRF
Facility	10	CRF	14	CRF	14	SVM	16	CRF
Vehicle	3	SVM	3	SVM	3	SVM	3	SVM
Weapon	3	SVM	3	SVM	3	SVM	3	SVM

Observations

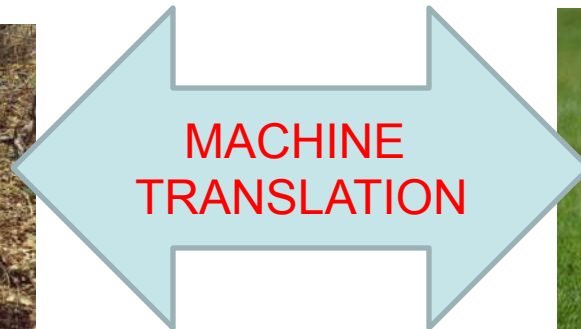
- SVMs and CRFs performances are very comparable, whereas ME performed very poorly
- When the results are compared per class, it has been observed that SVMs and CRFs lead to different results
- Best results are obtained when a combination-based approach is used and each classifier uses the ML technique which best fits the specific NE class

NER Integration in MT

- NE are a type of Multiword Expression (MWE)
- What are MWE?

MWE Definition

MWEs are a “key-problem” for the development of high-quality NLP applications



“kick the bucket” ??

MWEs

- Collocations of words that statistically co-occur more than chance
- Their semantic content *might* bear more than the meaning borne out by the individual words
- There is a strong correlation between idiomaticity and compositionality, the more idiomatic an expression, the less compositional

Multiword Expressions

- MWEs = “idiosyncratic interpretations that cross word boundaries (or spaces) or institutionalized phrases”
- Enormous number of them
 - In WordNet 3.0 (Fellbaum 1999), for example, ~40% of the entries are MWE
 - Specialized domain vocabulary, such as terminology, overwhelmingly consists of MWEs
- Problem for NLP
 - Compositionality versus Words-with-spaces (or dashes)

Compositionality vs. Words-with-spaces

- Compositionality problems
 - Over-generation
 - *Telephone booth*
 - * *Telephone cabinet*
 - Idiomaticity
 - *Kick the bucket*
- Words-with-spaces problems
 - Lack of flexibility
 - *Look up the <def> vs. Look the <def> up*
 - Lexical proliferation
 - Light verbs combos: *take a walk/hike/trip*
- How to account for variability
 - Segregate into different cases
 - Syntactic, Semantic, Inflectional variation

Research Questions

- What kind of information are we trying to model
 - All *types* of MWE without distinction with their morphological variants
 - *keep one's eyes peeled* is expanded into *keep her eyes peeled*
- How are we modeling it
 - Static Integration
 - Dynamic Integration
- Where are we modeling it in the SMT pipeline
 - Pre-alignment
 - Phrase Table

Two Integration Methods in MT

- Static Integration
 - Groups all words of an MWE into a single unit for Training, Test, Tune data (variant on segmentation)
 - Keep_one's_eyes_peeled
- Dynamic Integration
 - No preprocessing on the MWE till phrase table extraction (Alignments performed on words)
 - MWE detected in phrase table entries
 - Count freq weight added to phrase table probabilities creating a bias in the entry (don't break MWE)

DATA & Metrics

- Dictionary based MWE from WN 3.0 on English side ~79K MWE types
 - Pattern Forward Matching to detect MWE on tokens
 - Non adjacent and adjacent MWE
 - All types of MWE without regard to idiomaticity
- Open Domain, NW genre
- Training Data: 2.5M sentence pairs
- Test Data: MT08 813 English Sentences (500 MWE types corresponding to 900 MWE Tokens)
- Tuning Set: MT06 Data set
- Reference: Single Arabic Reference
- Evaluation Metric BLEU, NIST, TER

Experimental Conditions

- Baseline: Vanilla Moses System with no explicit MWE modeling
- Top 500 N-Grams (2-10 grams) using dynamic integration: only 10 overlap with WN MWE types
- Dynamic Integration of WN MWE
- Static Integration of WN MWE

Results

	Integration	BLEU
Baseline		30.49
Top 500 NGram	Dynamic	30.98
Dynamic WN MWE	Dynamic	31.07
Static WN MWE	Static	31.27

Observations

- Modeling MWE explicitly leads to gains
- Static Integration does the best
 - *Example:* the special envoy of the secretary-general will submit an oral report to the international security council rather than a *written report*
 - written report translated as tqryrA mktwbA vs Baseline ktb Altqryr (writing the re- port or book of report)
- Dynamic Integration can handle compositional MWE (Ngrams)
- However Sentence Level analysis reveals: different MWE require different Integration mechanisms
 - Dynamic Integration: who were then allowed to *take out* as many unsecured loans as they wanted (*take out* is dropped)

Nuanced MWE Integration into MT

- What
 - More Nuanced: Studying the different types of MWE separately
 - English to Arabic
- Where
 - Experimenting with different integration places
 - For Dynamic integration we align with underscores, then remove them prior to phrase extraction (measuring impact on WA)
- How
 - Similar Integration methods plus combination hybrids depending on MWE type

Data Characteristics

	MWE Categories	Types	Tokens
VAA	WN: FE, VPC, VNC, LVC	2537	186741
NNC	WN: NNC	7057	298286
NE	WN+SNER: NE (including Person)	74130	274724
NEP	WN+SNER: NE Person	26473	76656

MWE comprise 5% of the tokens corresponding to 43% of the types

Bleu Scores on English to Arabic

	Static	Dynamic (No underscores in alignment)
Baseline	38.09	38.24
VAA+NNC+NE (WN)	38.65	39.07
VAA+NNC+NE	35.9	38.95
VAA	39.06/+0.97	38.79
NNC	38.68	38.94
NE	36.41	39.41/+1.17
NEP	38.16	39.17

- MWE integration has a positive impact on SMT
- VAA seems to favor Static Integration
- NEs definitely favor Dynamic Integration
- NNC seem indifferent

Combining SI and DI for different MWEs

	Hybrid Results
<i>Baseline</i>	<i>SI: 38.09/DI: 38.24</i>
<i>All</i>	<i>SI: 35.9/DI: 38.95</i>
SI_VAA+DI_NE	38.73 (SI: +2.83/DI: -0.22)
SI_VAA+DI_NNC+DI_NE	38.53 (SI: +2.63/DI: -0.42)
SI_VAA+DI_VAA+DI_NE	39.51 (SI: +3.41/DI: +0.56)
SI_VAA+DI_VAA+DI_NNC+DI_NE	38.83 (SI: +2.93/DI: -0.12)

DI Indicates having a non-zero feature value for MWEs attested in Phrase Table

Integrating VAA Statically/dynamically and NE Dynamically yields the best results

Conclusions

- Tailoring the modeling to fit the data is a good thing (in this case we customized to Arabic morphology)
- When using an enabling technology understanding the underlying data allows for better integration
- Perform combination with a nuanced purpose

Thank You