HLT & NLP within the Arabic world:
Arabic Language and local languages processing
Status Updates and Prospects

WORKSHOP DATE: SATURDAY 31st May 2008

Workshop chair

Khalid Choukri
(ELRA/ELDA, France)

Workshop Co-chairs

Mona Diab,
(Columbia University, USA)
Bente Maegaard
(CST, University of Copenhagen, Denmark)
Paolo Rosso,
(Universidad Politécnica Valencia, Spain)
Abdelhadi Soudi
(ENIM, rabat, Morocco)
Ali Farghaly,
(Oracle USA and Monterey Institute of International Studies)

Program and Scientific Committee:

Ken Beesley , Xerox Research Centre Europe, France

Malek Boualem , France Telecom Orange Labs (France)

Tim Buckwalter , University of Maryland, (USA)

Violetta Cavalli-Sforza, San Francisco State University (USA)

Achraf Chalabi , Sakhr (Egypt)

Khalid Choukri, ELRA/ELDA (France)

Christopher Cieri, Linguistic Data Consortium, Philadelphia, (USA)

Fathi Debili, CELLMA - ENS LSH Lyon (France)

Mona Diab, Columbia University, (USA)

Joseph Dichy, Lyon -2 university Lyon (France)

Everhard Ditters, University of Nijmegen (The Netherlands)

Khaled Elghamry, (University of Florida, USA)

Ossama Emam, IBM (Egypt)

Ali Farghaly, Oracle USA and Monterey Institute of International Studies (USA),

Abdelkader Fassi-Fehri, University of Newcastle upon Tyne, (UK)

Gregory Grefenstette, LIC2M/CEA-LIST, (France)

Ahmed Guessoum, University of Sharjah, (UAE)

Nizar Habash, Columbia University, (USA)

Mohamed Hassoun, ENSIB, Lyon (France)

Steven Krauwer, ELSNET and Utrecht University Utrecht (The Netherlands)

Mohamed Maamouri, LDC, University of Pennsylvania, (USA)

Bente Maegaard, CST, University of Copenhagen, (Denmark)

Chafic Mokbel, University of Balamand (Lebanon)

Abdelhak Mouradi, ENSIAS (Morocco)

Owen Rambow, Columbia University, (USA)

Mohsen Rashwan, RDI (Egypt)

Horacio Rodríguez, Universitat Politécnica Catalunya, (Spain)

Mike Rosner, University of Malta, (Malta)

Paolo Rosso, Universidad Politécnica Valencia, (Spain)

Abdelhadi Soudi, ENIM (Morocco)

Mustafa Yassen, Amman University Amman (Jordan)

Introduction to the workshop

This Workshop, a satellite event to LREC 2008 main conference, intends to add value to the issues addressed during the main conference (Human Language Technologies (HLT) & Natural Language Processing (NLP)) and enhance the work carried out at different places to process Arabic language(s) and more generally Semitic languages and other local and foreign languages spoken in the region. It was a challenge to launch, once again, a workshop dedicated to Arabic. In some of the statements made at previous similar event by the organizers (e.g. LREC 2002 workshop on Arabic), it was clearly asserted that the ultimate goal would be that such papers integrate the main conference.

Despite this we felt the need to have another event specifically on arabic and local languages. Assuming this will bring together people who are actively involved in Arabic Written and Spoken language processing in a mono- or cross/multilingual context, and give them an opportunity to update the community through reports on completed and ongoing work as well as on the availability of LRs, evaluation protocols and campaigns, products and core technologies (in particular open source ones). This should enable the participants to develop a common view on where we stand with respect to these particular set of languages and to foster the discussion of the future of this research area. Particular attention will be paid to activities involving technologies such as Machine Translation, Cross-Lingual Information Retrieval/extraction, Summarization, Speech to text transcriptions, etc., and languages such as Arabic varieties, Amazigh, Amharic, Hebrew, Maltese, and other local languages. Evaluation methodologies and resources for evaluation of HLT are also a main focus.

The number of submissions received (about 23) shows how active the language technology community is with respect to processing Arabic. It also shows that local languages are not attracting the efforts they desserve but also that other specialised events focus on such topic in a better way. Only one paper deals with Amazigh databases. The submissins received show also that most of the themes mentioned in the Call for Papers are addressed even if a large part of the papers adress issues related to fundamental issues of arabc processing, that is morphology and syntax analysis, vowelisation, treebank, etc.

It is clear from the various projects that Arabic has become a major language for HLT. During this workshop we will emphasize the need to focus on specific issues that would help citizens living in Arabic countries to have access to information and technologies in their mother tongues and therefore discuss requirements to customize existing technologies for pairs of languages e.g. English to Arabic, Amazigh, etc. A particular stress will be put on tools, technologies, resources that tackle colloquial Arabic and other local languages such as Amazigh.

We expect to identify problems of common interest, and possible mechanisms to move towards solutions, such as sharing of resources, tools, standards, sharing and dissemination of information and expertise, adoption of current best practices, setting up joint projects and technology transfer mechanisms, etc.

By bringing together players in the Arabic NLP field, we would like to follow activities discussed at similar workshops (e.g. LREC2002) but also at the NEMLAR conference on Arabic Language (2004, Cairo Egypt), the workshop on Arabic NLP (Fez, April, 2007, http://www.dsic.upv.es/~prosso/workshopAECI_ArabicNLP.pdf) as well as work carried out in projects such as NET-DC (http://www.elda.org/article45.html), NEMLAR (www.nemlar.org) or the LDC project on the "Less Commonly Taught Languages" (http://projects.ldc.upenn.edu/LCTL/). The objective is also to introduce activities that will be launched shortly within the MEDAR project (the follow-up of NEMLAR project under FP7 of the European Commission). Among the crucial issues that require particular attention is the construction/update of a broadly supported Roadmap for these languages in relationship with Multilinguality and Evaluation of HLTs.

WORKSHOP PROGRAMME

Automatic versus interactive analysis for the massive vowelization, tagging and lemmatization of Arabic
Fathi Debili, Zied Ben Tahar,
LLACAN, INALCO, CNRS, France and Emna Souissi, ESSTT, Tunisia

Prague Arabic Dependency Treebank: A Word on the Million Words

Otakar Smrz, Viktor Bielicky, Iveta Kourilova, Jakub Kracmar, Jan Hajic, Petr Zemanek
Institute of Formal and Applied Linguistics, Charles University in Prague, Czech Republic

Arabic Named Entity Recognition using Conditional Random Fields
Yassine Benajiba and Paolo Rosso,
Natural Language Engineering Lab. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Spain.

Can the building of corpus-based Arabic concordances with AraConc and DIINAR.1 tackle the issue of Arabic polyglossia?
Joseph Dichy and Ramzi Abbès,
Université Lumière-Lyon 2 and ICAR (CNRS-Lyon 2)

Amazigh Data Base
El Mehdi IAZZI, Mohamed OUTAHAJALA,
Institut Royal de la Culture Amazigh, Rabat, Morocco

Building an Arabic Morphological Analyzer as part of an Open Arabic NLP Platform
Lahsen Abouenour (*), Said El Hassani(**), Tawfiq Yazidy (**), Karim Bouzouba(*), Abdelfattah Hamdani(**)
(*) Mohammadia School of Engineers, (**) Institute for Studies and Research on Arabization, Rabat, Morocco

Morpho-syntactic tagging system for Arabic texts
A. Yousfi , A. El jihad, and L. Aouragh,
IERA (Institute for Studies and Research on Arabization), Rabat Morocco

Guidelines for Annotation of Arabic Dialectness
Nizar Habash, Owen Rambow, Mona Diab and Reem Kanjawi-Faraj
Center for Computational Learning Systems, Columbia University, New York, NY, USA

Information retrieval in Arabic language
Malek Boualem (*), Ramzi Abbes (**)
(*) France Télécom Orange Labs, France; (**) Lyon 2 University / ICAR-CNRS, France

Memory-Based Vocalization of Arabic
Sandra Kübler, Emad Mohamed
Indiana University, Department of Linguistics, Bloomington, IN-47405, USA

Towards a human-machine spoken dialogue in Arabic
Younes Bahou, Lamia Hadrich Belguith, and Abdelmajid BEN HAMADOU
LARIS - MIRACL Laboratory, Faculty of Economic Sciences and Management of Sfax, Sfax, Tunisia

Methods for porting NL-based restricted e-commerce systems into other languages
Najeh Hajlaoui (*), Daoud Maher Daoud (**), Christian Boitet (*)
(*)GETALP, LIG,Université Joseph Fourier, Grenoble, France
(**) Amman University, Amman Jordan

Automatic Pronunciation Dictionary Toolkit for Arabic
Hussein Hiyassat(*), Mustafa Yaseen(**), Nihad Arabiat(***)
(*) e-Prucurment Project, UNDP, (**) Amman University, (***) Ministry of Education ; Amman, Jordan

Broadcast News Transcription Baseline System using the NEMLAR database
R. Bayeh (,**), C. Mokbel (**), G. Chollet (*)*
(*) TELECOM-ParisTech, CNRS-LTCI UMR-5141, Paris , France; (**) University of Balamand, Tripoli, Lebanon

Arabic-English translation improvement by target-side neural network language modeling

Maxim Khalilov(*), José A. R. Fonollosa(*), F. Zamora-Martínez(**), María J. Castro-Bleda(**), S. España-Boquera(**)
(*) Centre de Recerca TALP, Universitat Politècnica de Catalunya Barcelona, Spain; (**) Dep. de Lenguajes y Sistemas Informáticos, Universidad Politécnica de Valencia, Valencia, Spain

Language modeling for local and Modern Standard Arabic

Ilana Heintz, Chris Brew
Department of Linguistics, Ohio State University, Columbus, USA

Towards a syntactic lexicon of Arabic Verbs

Noureddine LOUKIL, Kais HADDAR, Abdelmajid BEN HAMADOU
Institut Supérieur d'Informatique et Multimédia de Sfax, Tunisie

Automatic Morphological Rule Induction for Arabic

Ahmad Hany Hossny (*), Khaled Shaalan (**), Aly Fahmy (*)
(*) Faculty of Computers and Information, Cairo University, Egypt
(**) Faculty of Informatics , The British University in Dubai, Dubai, UAE

ABSTRACTS

Automatic versus interactive analysis for the massive vowelization, tagging and lemmatization of Arabic

Fathi Debili, Zied Ben Tahar,
LLACAN, INALCO, CNRS, France and Emna Souissi, ESSTT, Tunisia

How could we produce annotated texts massively with optimal efficiency, reproducibility and cost? Instead of correcting the output of the automatic analysis with dedicated tools, as suggested currently, we found it more advisable to use interactive tools for analysis, where manual editing is fed in real time into automatic analysis. We address the issue of evaluating these tools, along with their performance in terms of linguistic ergonomy, and propose a metric for calculating the cost of editing as a number of keystrokes and mouse clicks. By way of a simple protocol addressing Arabic vowelization, tagging and lemmatization, we discover that, surprisingly, the best interactive performance of a system is not always correlated to its best automatic performance. In other words, the most performing automatic linguistic behavior of a system does not always yield the best interactive behavior, when manual editing is involved.

Prague Arabic Dependency Treebank: A Word on the Million Words

Otakar Smrz, Viktor Bielicky, Iveta Kourilova, Jakub Kracmar, Jan Hajic, Petr Zemanek
Institute of Formal and Applied Linguistics, Charles University in Prague, Czech Republic

Prague Arabic Dependency Treebank (PADT) consists of refined multi-level linguistic annotations over the language of Modern Written Arabic. The kind of morphological and syntactic information comprised in PADT differs considerably from that of the Penn Arabic Treebank (PATB). This paper explores the possibility to merge both of these treebanks into a uniform resource that would exceed the existing ones in the level of linguistic detail, accuracy, and quantity. It overviews the character of PADT and its motivations, and reports on converting and enhancing the PATB data. The merged, rule-checked and revised annotations, which amount to over one million words, as well as the open-source computational tools developed in the project are considered for publication this year.

Arabic Named Entity Recognition using Conditional Random Fields

Yassine Benajiba and Paolo Rosso,
Natural Language Engineering Lab. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Spain.

The Named Entity Recognition (NER) task consists in determining and classifying proper names within an open-domain text. This Natural Language Processing task proved to be harder for languages with a complex morphology such as the Arabic language. NER was also proved to help Natural Language Processing tasks such as Machine Translation, Information Retrieval and Question Answering to obtain a higher performance. In our previous works we have presented the first and the second version of ANERsys: an Arabic Named Entity Recognition system, whose performance we have succeeded to improve by more than 10 points, from the first to the second version, by adopting a different architecture and using additional information such as Part-Of-Speech tags and Base Phrase Chunks. In this paper, we present a further attempt to enhance the accuracy of ANERsys by changing the probabilistic model from Maximum Entropy to Conditional Random Fields which helped to improve the results significantly.

Can the building of corpus-based Arabic concordances with AraConc and DIINAR.1 tackle the issue of Arabic polyglossia?

Joseph Dichy and Ramzi Abbès,
Université Lumière-Lyon 2 and ICAR (CNRS-Lyon 2)

Research and development in Arabic NLP has nothing to gain from an oversimplified view of the Arabic language, considering the high level of linguistic variation commonly observed in the Arabic language as a whole, including Arabic vernaculars (or ‘Dialects’). The real challenge is both to keep hold of the current sate of knowledge in the description of linguistic variation – complex as it may seem, it is in itself a simplified representation when compared to real-world language use –, and at the same time, to seek efficient formalized approaches that allow software applications to operate at a sufficient level of granularity.

In this paper, we will try and show how a reasonable level of granularity and of subsequent feasibility can be reached through a balance between the complexity of Arabic corpus data and present-day tools that have been originally devised for Modern Standard Arabic (MSA). Section 1 introduces the complex system of the competence of communication in Arabic (Arabic polyglossia). Section 2 suggests a mapping of what could become, in the future, a real-world hyper-base of corpora in the Arabic language. Section 3 is an endeavour at contributing to the above challenge, through the use of tools initially devised for MSA. These are the DIINAR.1 knowledge database (*DIctionnaire INformatisé de l’ARabe, version 1*) and the last born of the analyzers based on it, the AraConc concordance software, which has been devised to extract concordances and frequency lists in Modern Standard Arabic. Two types of actual results are made available: statistical results and concordances. Examples of how these tools can meet linguistic variation in Arabic, and begin to evolve towards a new generation of tools are given.

Amazigh Data Base

El Mehdi IAZZI, Mohamed OUTAHAJALA,
Institut Royal de la Culture Amazigh, Rabat, Morocco

This paper focuses on the linguistic and computational aspects of a project which the Royal Institute for Amazigh Culture is carrying out. The project deals with the elaboration of an application that will help in collecting and accessing Amazigh words. The objective is to present some aspects of the Moroccan Amazigh data base project, its structure and the processing of variations within the framework of the linguistic norm elaborated in Morocco. The application modelling and computing tools used are also presented here. In order to optimise its use, the data base of the Amazigh language is conceived from the beginning in such a way as to provide all the necessary information for each entry. Furthermore, this information allows interrogating the data base from different angles: the classification of the glossary by domains (for example, agricultural glossary, handcraft glossary….), the derivational families such as words derived from the same root, Arabic French and English equivalent words, etc.

Building an Arabic Morphological Analyzer as part of an Open Arabic NLP Platform

Lahsen Abouenour (*), Said El Hassani(**), Tawfiq Yazidy (**), Karim Bouzouba(*), Abdelfattah Hamdani(**)
(*) Mohammadia School of Engineers, (**) Institute for Studies and Research on Arabization, Rabat, Morocco

For many reasons (growth of Arabic Internet, greater interest on Arabic Media, etc.), Arabic NLP is facing many challenges. To contribute in answering some of them, we present in this paper an integrated and open Arabic NLP platform. This platform is dedicated to the development of many kinds of Arabic NLP applications. In addition of its openness, this platform is aimed to respect criteria such as standardization, flexibility and reusability. As a first step for the development of the platform, we present also its morphological layer with a focus on Arabic nouns. This analyzer is mainly based on a new classification of the Arabic nouns and provides useful information for other layers (syntax and semantics). Experiments done on selected corpora are very encouraging and the sketched architecture leads to many other interesting future works.

Morpho-syntactic tagging system for Arabic texts

A. Yousfi , A. El jihad, and L. Aouragh,
IERA (Institute for Studies and Research on Arabization), Rabat Morocco

Text tagging is a very important tool for various applications in natural language processing, namely the morphological and syntactic analysis of texts, indexation and information retrieval, "vocalization" of Arabic texts, and probabilistic language model (n-class model). However, these systems based on the lexemes of limited size, are unable to treat unknown words consequently.

To overcome this problem, we developed in this paper, a new system based on the patterns of unknown words and the Hidden Markov Model (HMM). The experiments are carried out in the set of labeled texts, the set of 3800 patterns, and the 52 tags of morpho-syntactic nature, to estimate the parameters of the new model HMM.

Guidelines for Annotation of Arabic Dialectness

Nizar Habash, Owen Rambow, Mona Diab and Reem Kanjawi-Faraj
Center for Computational Learning Systems, Columbia University, New York, NY, USA

The Arabic language is a collection of dialects with phonological, morphological, lexical, and syntactic differences comparable to those among the Romance languages. However, throughout the Arab world, the standard written language is the same, Modern Standard Arabic (MSA). MSA is used in some official spoken communication such as newscasts, parliamentary debates, etc. MSA is based on Classical Arabic and is itself not a native spoken language. Other forms of Arabic (generally referred to as “dialects” of MSA) are what people use for daily spoken communication. In this paper, we focus on the issue of creating standard annotation guidelines identifying dialect switching between MSA and at least one dialect in written text. These guidelines can form the basis for the annotation of large collections of data that will be used for training and testing automatic approaches to dialect identification and automatic processing of Arabic which exhibits dialect switching. Dialect identification can be useful for dialect normalization or automatic conversion of dialects to MSA. We report on some initial annotation experiments: we discuss statistical distributions of labels in a small corpus (~19K words) that we annotated according to the guidelines and present inter-annotator agreement results.

Information retrieval in Arabic language

Malek Boualem (*), Ramzi Abbes (**)
(*) France Télécom Orange Labs, France; (**) Lyon 2 University / ICAR-CNRS, France

Web search engines provide quite good results for Latin characters-based languages. However, they still show many weaknesses when searching in other languages such as Arabic. This paper discusses a qualitative analysis of information retrieval in Arabic, highlighting some of the numerous limitations of available search engines, mainly when they are not properly adapted to the Arabic language features. To support our analysis we present some results based on thorough observations about various Arabic linguistic phenomena. To validate these observations, we mainly have tested the Google search engine. Arabic information retrieval still faces many difficulties due to the Arabic linguistic features, especially its complex morphology and the absence of vowels in available documents and texts. These

specificities often cause significant dissymmetry between the indexation process and the query analysis. We present in this paper some of the morphological constraints of Arabic language and we show through experimental tests how search engines deal with them. Finally this paper clearly states that information retrieval in Arabic language will never succeed without including language processing tools at all the linguistic levels (lexical, syntactic and semantic).

Memory-Based Vocalization of Arabic

Sandra Kübler, Emad Mohamed
Indiana University, Department of Linguistics, Bloomington, IN-47405, USA

The problem of vocalization, or diacritization, is essential to many tasks in Arabic NLP. Arabic is generally written without the short vowels, which leads to one written form having several pronunciations with each pronunciation carrying its own meaning(s). In the experiments reported here, we define vocalization as a classification problem in which we decide for each character in the unvocalized word whether it is followed by a short vowel. We investigate the importance of different types of context. Our results show that the combination of using memory-based learning with only a word internal context leads to a word error rate of 6.64% on newswire text. However, if punctuation and numbers are excluded from vocalization, the best results are reached by including the left context. On the data set by Zitouni et al. (2006), our best performing system reached a word error rate of 17.5%.

Towards a human-machine spoken dialogue in Arabic

Younes Bahou, Lamia Hadrich Belguith, and Abdelmajid BEN HAMADOU
LARIS - MIRACL Laboratory, Faculty of Economic Sciences and Management of Sfax, Sfax, Tunisia

This work is a part of automatic spontaneous Arabic speech understanding. We propose in this paper an analysis method guided by semantics and based on the frame grammar formalism. This method allows representing meaningful oral utterances in semantic frame forms. It has the advantage of being robust when faced to analysis problems due to the spontaneity of interaction and the speech recognition limits. In this paper, we present our system of automatic Arabic speech understanding applied to the domain of Tunisian railway information. The understanding module of this system is based on the proposed method.

Methods for porting NL-based restricted e-commerce systems into other languages

Najeh Hajlaoui (*), Daoud Maher Daoud (**), Christian Boitet (*)
(*)GETALP, LIG,Université Joseph Fourier, Grenoble, France
(**) Amman University, Amman Jordan

Multilingualizing systems handling content is an important but difficult problem. As a manifestation of this difficulty, very few multilingual services are available today. The process of multilingualization depends on the translational situation: types and level of possible accesses, available resources, and linguistic competences of participants involved in the multilingualization of an application. Several strategies of multilingualization are then possible (by translation, by internal or external localization etc.). We present a real case of linguistic porting (from Arabic to French) of an e-commerce application deployed in Jordan, using spontaneous SMS in Arabic for buying and selling second-hand cars. Despite the distance between Arabic and French, the localization methods used give good results because of the proximity of the two sublanguages of Arabic and French in this restricted domain.

Automatic Pronunciation Dictionary Toolkit for Arabic

Hussein Hiyassat(*), Mustafa Yaseen(**), Nihad Arabiat(***)
(*) e-Prucurment Project, UNDP, (**) Amman University, (***) Ministry of Education ; Amman, Jordan

Speech Recognizers are commercially available from different vendors. Along with this increased availability comes the demand for recognizers in many different languages that often were not focused on the speech recognition research. So far, Arabic language is one of those languages. With the increasing role of computers in our lives, there is a demand to communicate with them naturally. Speech processing by computer provides one vehicle for natural communication between man and machine. In this paper, a novel approach for implementing Arabic isolated speech recognition is described. The first SPHINX-IV-based Arabic recognizer is introduced and an automatic toolkit is proposed, which is capable of producing pronunciation dictionary (PD) for both Holly Qura'an and Arabic digits corpus ADC. ADC corpora were tested and evaluated accuracy of 99.213% and WER is 0.787% is obtained.

Broadcast News Transcription Baseline System using the NEMLAR database

R. Bayeh (,**), C. Mokbel (**), G. Chollet (*)*
(*) TELECOM-ParisTech, CNRS-LTCI UMR-5141, Paris , France; (**) University of Balamand, Tripoli, Lebanon

This paper describes one of the first uses of the NEMLAR Arabic Broadcast News Speech Corpus (BNSC) for the creation of an automatic speech recognizer (ASR) for Arabic Broadcast News (BN). Different parameterization settings, types of acoustic models, various language models and testing schemes are presented for the creation of a baseline system for Modern Standard Arabic using the NEMLAR BNSC database. To port this system to dialects, a certain amount of dialectal data is required. Due to the absence of such resources and the use of other languages in dialectal speech, techniques for the creation of cross-lingual models using the baseline system are investigated. Certain techniques that have given promising results in previous experiments are proposed. These techniques, which would be helpful in developing a cross-dialectal speech recognition system, have been, due to the use of Maghrebian/Levantine dialects which make use of French, experimented in a cross-lingual Arabic-French frame. Although a lot of work remains to be accomplished, the current results are very encouraging.

Arabic-English translation improvement by target-side neural network language modeling

Maxim Khalilov(*), José A. R. Fonollosa(*), F. Zamora-Martínez(**), María J. Castro-Bleda(**), S. España-Boquera(**)
(*) Centre de Recerca TALP, Universitat Politècnica de Catalunya Barcelona, Spain; (**) Dep. de Lenguajes y Sistemas Informáticos, Universidad Politécnica de Valencia, Valencia, Spain

The quality of translation, produced by Statistical Machine Translation (SMT) systems, crucially depends on generalization, provided by the statistical models involved into translation process. In this study we present the n-gram based translation system (i.e. the UPC SMT system), enhanced with a continuous space language model (LM), estimated with a neural network (NN). In the framework of the study, we use NN LM on the rescoring step, reevaluating the N-best list of complete translations hypothesis. Different word history length included in the model (n-gram order) and distinct continuous space representation (i.e. including words, appearing in the training corpus more than k times) are considered in the paper. We report result for an Arabic-English translation task, improving Arabic-English translation accuracy by better target language model representation in contrast with the state-of-the-art approach. The experimental results are evaluated by means of automatic evaluation metrics correlated with fluency and adequacy of the generated translations.

Language modeling for local and Modern Standard Arabic

Ilana Heintz, Chris Brew
Department of Linguistics, Ohio State University, Columbus, USA

We propose a Finite State Machine framework for Arabic Language Modeling. The framework provides several decompositions per word based on the forms of possible stems. The statistical modeling is responsible for ranking the most plausible (prefix)-stem-(suffix) sequences higher than the less plausible decompositions. In addition to being useful for Modern Standard Arabic, we show that the framework is easily applied to colloquial Arabic, which suffers from low amounts of text data for use in Natural Language Processing.

Towards a syntactic lexicon of Arabic Verbs

Noureddine LOUKIL, Kais HADDAR, Abdelmajid BEN HAMADOU
Institut Supérieur d'Informatique et Multimédia de Sfax, Tunisie

This paper presents the modelling of an extensional syntactic lexicon for verbal entities in Arabic, based on the initiative for lexical resources normalisation LMF (Lexical Markup Framework). The specific syntactic behaviors for verbs in Arabic language are identified and presented with examples. Each verbal entry is specified with a list of accepted syntactic patterns describing the set of accepted arguments with their different constraints. The syntactic extension of LMF and the XML structure of lexical entries are presented in accordance with the LMF normative initiative. This lexicon would be very useful for NLP community because it enables comfortable use in applications due to its normalised representation.

Automatic Morphological Rule Induction for Arabic

Ahmad Hany Hossny (*), Khaled Shaalan (**), Aly Fahmy (*)
(*) Faculty of Computers and Information, Cairo University, Egypt
(**) Faculty of Informatics , The British University in Dubai, Dubai, UAE

In this paper, we introduce an algorithm for morphological rule induction using meta-rules for Arabic morphology based on inductive logic programming. The processing resources are a set of example pairs (stem and inflected form) with their feature vectors, either positive or negative, and the linguistic background knowledge from the Arabic morphological analysis domain. Each example pair has two words to be analyzed vocally into consonants and vowels. The algorithm applies two levels of mapping: between the vocal representation of the two words (stem, morphed) and between their feature vector. It differentiates between both mappings in order to accurately deduce which changes in the word structure led to which changes in its features. The paper also addresses the irregularity, productivity and model consistency issues. We have developed an Arabic morphological rule induction system (AMRIS). Successful evaluation has been performed and showed that the system performance results achieved were satisfactory.

PAPERS

Automatic *versus* interactive analysis for the massive vowelization, tagging and lemmatization of Arabic

Fathi Debili
LLACAN, INALCO, CNRS
7, rue Guy Môquet
94801 Villejuif cedex
France
fathi.debili@wanadoo.fr

Zied Ben Tahar
LLACAN, INALCO, CNRS
7, rue Guy Môquet
94801 Villejuif cedex
France
bentaharzied@gmail.com

Emna Souissi
ESSTT
5, Av Taha Hussein
Tunis
Tunisie
emna.souissi@planet.tn

Abstract

How could we produce annotated texts massively with optimal efficiency, reproducibility and cost? Instead of correcting the output of the automatic analysis with dedicated tools, as suggested currently, we found it more advisable to use interactive tools for analysis, where manual editing is fed in real time into automatic analysis. We address the issue of evaluating these tools, along with their performance in terms of linguistic ergonomy, and propose a metric for calculating the cost of editing as a number of keystrokes and mouse clicks. By way of a simple protocol addressing Arabic vowelization, tagging and lemmatization, we discover that, surprisingly, the best interactive performance of a system is not always correlated to its best automatic performance. In other words, the most performing automatic linguistic behavior of a system does not always yield the best interactive behavior, when manual editing is involved.

1 Introduction

Automatic analysis seems to have been present before interactive analysis, by which we mean a partially manual one. Automatic analysis has been the main concern of most researchers from the beginning, and it may well remain, for quite a long time, the goal we try to reach and the perfection we try to achieve. What we'll call « artisanal » handmade analysis has also been performed from early stages, with various aims, among which, the one of building annotated corpora for learning or evaluation. Even though researchers very quickly found about the real difficulties of manual analysis, they were late in applying steady efforts to it, guided by a demand for better performance and for larger coverage. In this context, annotation guides appeared, in order to make this process as reproducible as possible (Adda et al. 1999, Véronis, 1999, Abeillé and Clément, 2003). Later on, dedicated software tools were developed, incorporating some growing use of automatic analysis to serve the manual one (Habert, 2005). This paper follows this track and goes further along it. We deal with issues concerning the massive annotation of Arabic corpora, along with its manual verification and correction, in other words, the interactive morpho-grammatical analysis of Arabic.

2 High rates of ambiguity

Arabic words as found in texts, namely their inflected (simple or agglutinated) forms, which we'll call hyperforms, have a high rate of segmental, vocalic, case-related, lemmatic and grammatical ambiguity. Table 1 gives mean values from dictionary and text data. One dictionary has 66 million non-vowelled entries obtained by lexico-syntagmatic synthesis. Another one has 157 thousand entries from a corpus with 2 million tokens, which in its turn was used as text.

Ambiguity	Segmental	Vocalic and Case-related	Lemmatic	Grammatical
Dictionary $66x10^6$	1.08	2.17	1.68	2.99
Lexicon 157 031	1.26	6.40	2.65	9.16
Corpus $2x10^6$	1.32	7.84	3.66	10.76

Table 1: Rate of ambiguity in Arabic hyperforms

These values show that ambiguity rates are much higher in Arabic than, for instance, in French (Debili & al., 2002). Another more global measurement was performed. It is related to the rate of combined ambiguity, namely, when one puts together segmental, vocalic, case-related, lemmatic and grammatical ambiguities. Lexico-syntagmatic synthesis yields, for 500 thousand simple non-vowelled inflected Arabic forms, 305 million different simple and agglutinated forms, vowelled, lemmatized and tagged, corresponding to 66 million simple and agglutinated, non-vowelled forms. The ratio of 305 to 66 yields a mean ambiguity of about 4.6 different morpho-grammatical readings per entry. It amounts to 14.7 for the sub-lexicon with 157 thousand entries, and 16.7 for the corpus with 2 million tokens.

3 High costs for annotating and data entry

In Arabic, most letters (87% in definition, 77% in use) need, for vowelization, a diacritic sign with an entry cost of two keystrokes, the same as a trema in French. The entry of Arabic vowelled letters is therefore expensive, with an average of three strokes. Table 2 shows the average cost of one character in keystrokes for various corpora: French (673 thousand words), English (650 thousand words), vowelled Arabic (800 thousand words) and unvowelled Arabic (2 million words).

	Cost	Proportion of diacritic signs	Proportion in entry cost
English	1.00001	0.0005 %	0.001 %
French	1.003	3.51 %	3.84 %
Arabic (non vowelled)	1.037	-	-
Arabic (vowelled)	1.46	43.7 %	59.9 %

Table 2: Average cost of a character in keystrokes

According to the table, entering a N-character text costs N×1.00001 keystrokes if the text is in English, compared to N×1.003 if it is in French, N×1.037 if it is in non-vowelled Arabic, and N×1.46 if it is in vowelled Arabic. One should also consider the vowelization of a text previously entered is of no lesser cost than the entry of a vowelled version. Therefore, the vowelization of Arabic is a prohibitive expense, without special precautions.

These measurements are of course related to the state of the art and to the keyboards used for each of the three languages. They offer an *a posteriori* evaluation of current norms and standards, and may contribute to their confirmation or correction. But they also provide an opportunity for recognizing the difficulty of a massive construction of annotated corpora, and for introducing metrics that go beyond classical evaluation metrics for automatic analysis, namely, metrics for quantitative evaluation of interactive analysis, based on the cost of manual processing.

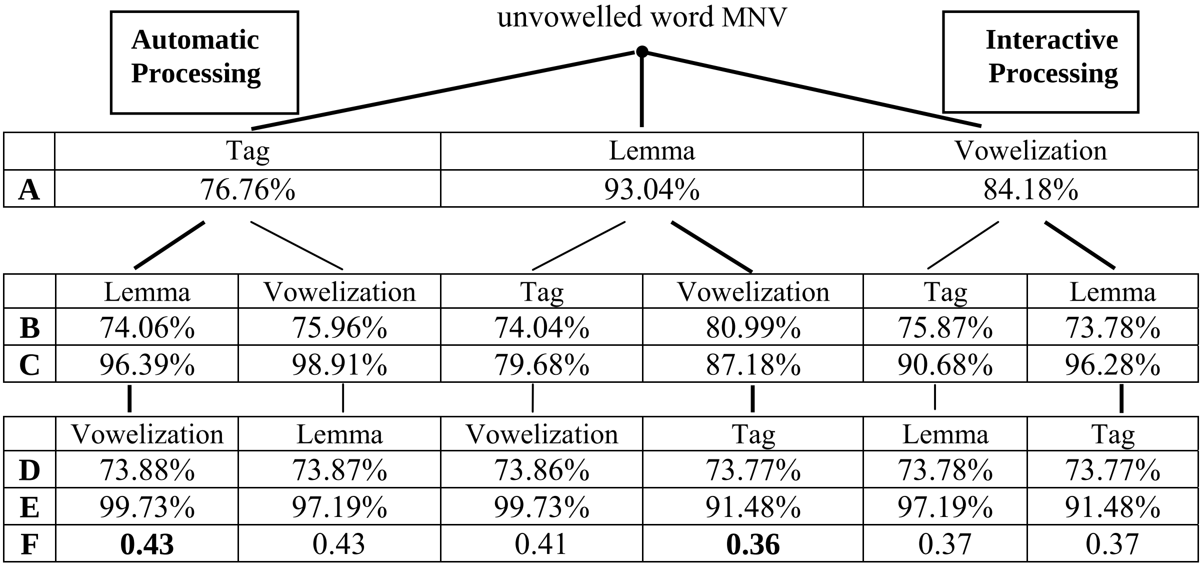
4 Interactive sequential annotation

Three annotations are required for each token (vowelization, lemma and tag). These annotations may interfere and have a dynamic influence on the order in which future annotations will be done. Therefore, six annotation protocols can be selected. The tree in Figure 1 shows these six possibilities. A seventh protocol corresponds to the case of independent, non-interferent choices.

Two protocols are of special interest to us: Tagging-Lemmatisation-Vowellisation (TLV, left on Figure 1) and Vowellisation-Lemmatisation-Tagging (VLT, right). The first one corresponds to automatic processing (the machine), and the second one to manual processing (human annotators). They are supposed to have better performances than the other ones.

But since rules interact with each other, we do not know *a priori* which of these sequences will yield the most performing automatic or interactive analysis. To measure this, we have elaborated and implemented the following experimental device.

From a corpus with 145 thousand hyperforms (all vowelled, lemmatized and tagged), we have extracted the relative frequencies:



	Tag	Lemma	Vowelization
A	76.76%	93.04%	84.18%

	Lemma	Vowelization	Tag	Vowelization	Tag	Lemma
B	74.06%	75.96%	74.04%	80.99%	75.87%	73.78%
C	96.39%	98.91%	79.68%	87.18%	90.68%	96.28%

	Vowelization	Lemma	Vowelization	Tag	Lemma	Tag
D	73.88%	73.87%	73.86%	73.77%	73.78%	73.77%
E	99.73%	97.19%	99.73%	91.48%	97.19%	91.48%
F	**0.43**	0.43	0.41	**0.36**	0.37	0.37

Line **A**: Automatic performance, Application of rules f(T|MNV), f(L|MNV), f(V|MNV).
Line **B**: Automatic performance, Application of rules f(L|MNV, T), f(V|MNV, T), f(T|MNV, L), f(V|MNV, L), f(T|MNV, V), f(L|MNV, V).
Line **C**: Interactive performance, Application of rules f(L|MNV, T), f(V|MNV, T), f(T|MNV, L), f(V|MNV, L), f(T|MNV, V), f(L|MNV, V). Here, in conditions | MNV, *y*), *y* is correct.
Line **D**: Automatic performance, Application of rules f(V|MNV, T, L), f(L|MNV, T, V), f(V|MNV, L, T), f(T|MNV, L, V), f(L|MNV, V, T), f(T|MNV, V, L).
Line **E**: Interactive performance, Application of rules f(V|MNV, T, L), f(L|MNV, T, V), f(V|MNV, L, T), f(T|MNV, L, V), f(L|MNV, V, T), f(T|MNV, V, L). Here, in | MNV, *y*, *z*), *y* and *z* are correct.
Line **F**: Cost of correction
MNV : unvowelled word or hyperform ; T : tag ; V : vowelization ; L : lemma

Figure 1: Performance of automatic and interactive analysis

f(tag | unvowelled word),
f(lemma | unvowelled word),
f(vowelization | unvowelled word),
f(lemma | unvowelled word, tag), etc., (see legend of Figure 1).

These frequencies were used as unary rules applied in cascade along the six protocols. We calculated their performances and, in a retrospective way, the costs that would have been involved in manual corrections.

Figure 1 displays these results. Lines A, B and D show tagging, lemmatization and vowelization performances using rule sequences f(*x*| unvowelled word), f(*x*| unvowelled word, *y*) and f(*x*| unvowelled word, *y*, *z*), with *x*, *y*, *z* = T, L or V. Lines C and E show the performance with manual correction for *y* and *z*. F shows average cost per word for manual validation (zero cost) or mouse pointing of T, L or V according to context. This cost varies like the rank of the pointed solution in the list of potential ones, after the elimination of some of them.

We find that correction costs are all different. Besides, these costs are not correlated with automatic analysis performance, as could have been expected.

A spatial distribution cuts Figure 1 into two regions according to performances and costs (see also Figure 2). According to this distribution, one should not favour the most performing automatic process. Less performing one should be preferred in an interactive context.

The distribution also confirms the well-foundedness of *a priori* recommended approaches for automatic or interactive analysis.

5 Conclusion

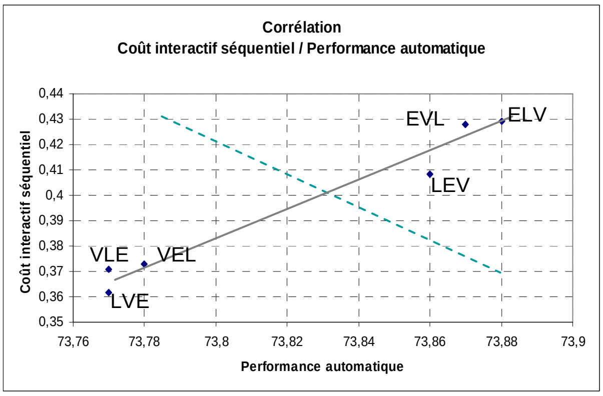


Figure 2: Correlation between Interactive cost and Automatic performance

The expected Cost-Performance correlation is better automatic performance corresponds to less interactive cost and dotted curve of tendency rather than a continuous line as observed in the experiment.

This diagram reproduces the results in lines D and F in Figure 1. It shows that *in a given performance range, the best automatic protocol will not necessarily, in case of manual intervention, yield the best interactive one.*

In other words, and drawing attention to the local aspect of our remarks, we discover that the most performing autonomous behaviour does not always yield, in case of interaction, the most performing cooperative behaviour. This empirical result is quite surprising. It has become the most important discovery to us during the course of our work, beyond our initial purpose of interactive analysis of Arabic corpora.

From a methodological point of view, it challenges current strategies in massive annotating, which take for granted that the best automatic analyser will be the best choice for interactive processing. We think that this is only true when the automatic performance is above a certain threshold. However, further testing is required to prove it. Below this threshold, the correlation between *best automatic* and *best interactive* performance is not valid.

Evaluation of interactive annotation proves that, while it is crucial to improve automatic performance, it is also useful to provide not one, but several behaviours to such analysers, to give them the best chances to interact harmoniously with human annotators.

Acknowledgement

Jean-Baptiste Berthelin, translated this article, and is always willing to discuss such topics.

This work was initiated within project EurADic (Action Technolangue du Ministère de la recherche), and goes on within project MUSCLE (6th PCRD).

References

Anne Abeillé, Lionel Clément (2003). *Annotation morpho-syntaxique. Les mots simples – Les mots composés. Corpus Le Monde*. Technical report, Paris 7.

Gilles Adda, Joseph Mariani, Patrick Paroubek, Martin Rajman, & Josette Lecomte (1999). L'action GRACE d'évaluation de l'assignation des parties du discours pour le français. *Langues*, 2(1).

Fathi Debili, Hadhémi Achour, Emna Souissi (2002). *La langue arabe et l'ordinateur : de l'étiquetage grammatical à la voyellation automatique*. Correspondances N°71, IRMC, Tunis, 10-26. http://www.irmcmaghreb.org/IMG/pdf/correspondances_71.pdf

Benoît Habert (2005). *Instruments et ressources électroniques pour le français*. Paris: Editions Ophrys.

Jean Véronis (1999). *Guide d'étiquetage Multitag*. Version 3.1, 6 novembre 1999.

Prague Arabic Dependency Treebank: A Word on the Million Words

Otakar Smrž Viktor Bielický Iveta Kouřilová Jakub Kráčmar Jan Hajič Petr Zemánek

Institute of Formal and Applied Linguistics, Charles University in Prague
Malostranské náměstí 25, Prague 1, 118 00, Czech Republic
`http://ufal.mff.cuni.cz/padt/`   `<padt@ufal.mff.cuni.cz>`

Abstract

Prague Arabic Dependency Treebank (PADT) consists of refined multi-level linguistic annotations over the language of Modern Written Arabic. The kind of morphological and syntactic information comprised in PADT differs considerably from that of the Penn Arabic Treebank (PATB). This paper overviews the character of PADT and its motivations, and reports on converting and enhancing the PATB data in order to be included into PADT. The merged, rule-checked and revised annotations, which amount to over one million words, as well as the open-source computational tools developed in the project are considered for publication this year.

1. Introduction

Prague Arabic Dependency Treebank (PADT) provides refined linguistic annotations inspired by the Functional Generative Description theory (Sgall et al., 1986; Hajičová and Sgall, 2003) and the Prague Dependency Treebank project (Hajič et al., 2006). The multi-level description scheme discerns functional morphology, analytical dependency syntax, and tectogrammatical representation of linguistic meaning. PADT is maintained by the Institute of Formal and Applied Linguistics, Charles University in Prague. The initial version of PADT (Hajič et al., 2004a) covered over one hundred thousand words of text.

PADT was included in the CoNLL 2006 and CoNLL 2007 Shared Task on dependency parsing (Nivre et al., 2007) or in other parsing experiments (Corston-Oliver et al., 2006). The morphological data and methodology of PADT were also used for training automatic taggers (Hajič et al., 2005). PADT is discussed in detail in (Žabokrtský and Smrž, 2003; Hajič et al., 2004b; Smrž, 2007b; Smrž and Hajič, 2008).

Penn Arabic Treebank (PATB) is the largest such resource for Modern Written Arabic that is annotated with structural morphological features, morph-oriented English glosses, and labelled phrase-structure syntactic trees in the predicate-argument style of the Penn Treebank (Marcus et al., 1993). PATB was developed at the Linguistic Data Consortium, University of Pennsylvania, and was published gradually in four major releases (Maamouri et al., 2004a, 2005a,b,c). The source texts are distributed also independently as part of the Arabic Gigaword (Graff, 2007).

PATB has been used mostly for the availability of morphological annotations and fully vocalized word forms. Processing the data with machine-learning techniques has resulted in a number of morphological taggers (Habash and Rambow, 2005; Smith et al., 2005; Hajič et al., 2005) and diacritizers (Nelken and Shieber, 2005; Zitouni et al., 2006; Habash and Rambow, 2007). The syntactic information was exploited in particular for parsing (Kulick et al., 2006), grammar extraction (Habash and Rambow, 2004), and automatic case assignment (Habash et al., 2007).

The PATB treebank is further described in (Maamouri and Bies, 2004; Buckwalter, 2004a; Maamouri et al., 2004b).

This paper explores the possibility to merge both of these treebanks into a uniform resource that would exceed the existing ones in the level of linguistic detail, accuracy, and quantity. While we advance the PADT style of annotations in this effort, we also largely benefit from the amount of disambiguated information available in PATB.

The new more than one-million-word treebank denoted as PADT 2.0 combines original Prague annotations with the transformed and enhanced Penn data. The preliminary contents of these components are enumerated in Table 1.

2. Functional Morphology

Due to the impact of PATB, the computational linguistics community is well aware of the Buckwalter Arabic Morphological Analyzer (Buckwalter, 2002, 2004b). This system can be characterized as following the lexical–incremental approach to morphology (Stump, 2001), implying that the only clue for discovering a word's morphosyntactic properties is through its explicit morphs and their supposed prototypical functions.

The functional view of language pursued in PADT requires, on the contrary, an inferential–realizational morphological model capable of more appropriate and deeper generalizations. ElixirFM (Smrž, 2007a,b) is the novel implementation that replaces any earlier functional approximations used in PADT, which were developed also thanks to the Buckwalter analyzer (Smrž and Pajas, 2004).

In order to illustrate the differences in the morphological description of PATB versus PADT, let us discuss a few examples. One disambiguated word in the PATB data might offer this information:

Form	`All~Asilokiy~apu`	اللاَّسِلْكِيَّةُ
Morph	`Al` + `lAsilokiy~` + `ap` + `u`	
Tag	`DET+ADJ+NSUFF_FEM_SG+CASE_DEF_NOM`	
Gloss	the + wireless / radio + [fem.sg.] + [def.nom.]	
Lemma	[`lAsilokiy~_1`]	لاَسِلْكِيّ
Root	*implicit in the lexicon*	

The entry spells out the full inflected word form in the Buckwalter transliteration, identifies its structure, and describes it with tags and glosses. It also provides the citation form of the lexeme that the word form represents. Other information relevant to the lexeme, like its derivational root, might be stored implicitly in the Buckwalter lexicon, but is not readily available in the treebank.

Data Set		Corpus	Functional Morphology			Dependency Syntax			Tectogrammatics		
		'words'	tokens	paras	docs	tokens	paras	docs	tokens	paras	docs
Prague	**AEP**	99360	**116717**	3006	327	**116717**	3006	327	**9690**	242	29
	EAT	48371	**55097**	1667	207	**55097**	1667	207	**13934**	436	58
	ASB	11881	**14254**	558	36	**14254**	558	36			
	NHR	21445	**25329**	426	34	**12613**	209	17			
	HYT	85683	**100537**	1782	204	**41855**	796	91	**5228**	106	10
	XIN	61500	**71548**	2389	321	**41716**	1429	196	**2042**	75	13
Penn	**1v3**	141515	**161217**	4790	628	**161217**	4790	628			
	2v2	140821	**163973**	2929	476	**163973**	2929	476			
	3v2	335250	**394466**	12445	589	**394466**	12445	589			
	4v1	149784	**178720**	5618	361						
Prague		328240	**383482**	9828	1129	**282252**	7665	874	**30894**	859	110
Penn		767370	**898376**	25782	2054	**719656**	20164	1693			
PADT 2.0		1095610	**1281858**	35610	3183	**1001908**	27829	2567	**30894**	859	110

Table 1: Expected contents of PADT 2.0. The Prague data sets AEP and EAT cover parts of the Arabic English Parallel News (Ma, 2004) and the full English-Arabic Treebank (Bies, 2006), while ASB, NHR, HYT, and XIN are selected from the Arabic Gigaword (Graff, 2007). The Penn data correspond to the parts and versions of PATB, modulo duplicate documents.

The corresponding PADT entry by ElixirFM would yield:

Form	`al-lA-silkIyaTu`	*al-lā-silkīyatu* اَللَّاسِلْكِيَّةُ
Morph	`al >| lA >| FiCL |< Iy |< aT |<< "u"`	
Tag	`A-----FS1D`	

Form	`lA-silkIy`	*lā-silkīy* لَاسِلْكِيّ
Morph	`lA >| FiCL |< Iy`	
Root	`"s l k"`	
Reflex	wireless, radio	
Class	adjective	

The interpretation is a little more formal. The lexeme of the given grammatical class and meaning is inflected in the parameters expressed by the tag—the adjective is inflected for feminine gender, singular number, nominative case, and definite state. Both the inflected word form and the citation form are explicit in their morphological structure—they are specified via the underlying template of morphs and the inherited root. Merging the template with the root produces the form in the ArabTEX notation, from which the orthographic string or its phonetic version can be generated.

In this very instance, both of the treebank entries seem more or less equivalent. With some other kinds of words, however, the PATB morphology systematically fails to determine many of the contextual and lexical parameters:

Form	`waOuxoraY`	وَأُخْرَى
Morph	`wa` + `OuxoraY`	
Tag	`CONJ+ADJ`	
Gloss	and + other / another / additional	

Lemma	[`OuxoraY_1`]	أُخْرَى

The word to analyze is in fact two lexical words, 'and' and 'other', joined in writing. There are two morphs and two tags, one for the conjunction, one for the adjective. There is yet only one explicit lemma. There are no details about the gender, number, case, or state of the adjective. Linguistically, though, the adjective is feminine singular with some possible, but not actual, ambiguity in case and state, and the lexeme's citation form is not as indicated.

The complete analysis in PADT would rather supply these individual tokens:

Form	`'u_hrY`	*ʾuḫrā* أُخْرَى	`wa`	*wa* وَ
Morph	`FuCLY |<< "u"`		`"wa"`	
Tag	`A-----FS1I`		`C---------`	

Form	`'A_har`	*ʾāḫar* آخَر	`wa`	*wa* وَ
Morph	`HACaL`		`"wa"`	
Root	`"' _h r"`		`"w"`	
Reflex	other, another		and	
Class	adjective		conjunction	

ElixirFM carefully designs the morphophonemic patterns of the templates, as well as the phonological rules hidden in the `>|` or `|<<` operators. This greatly simplifies the morphological rules proper, both inflectional and derivational. Inspired by functional programming in Haskell (Forsberg and Ranta, 2004), ElixirFM implements many generalizations of classical grammars (Fischer, 2002), and suggest even some new abstractions (Smrž, 2007b; compare the approach to patterns in Ryding, 2005; Yaghi and Yagi, 2004).

One would expect that the most underspecified words in a treebank might be those with weak morphology (Habash et al., 2007). In our final example, though, the problem lies elsewhere—how do we motivate the morphs, how do we define the tokens, how do we interpret the tags, and how do we ensure the uniformity of this information in all the data?

PATB does not separate the future marker *sa-* from an indicative verb, but does handle distinct tokens if the markers are the stand-alone *sawfa* or *lan*. Likewise, perhaps unintentionally, *li-* is not tokenized if followed by a jussive, but it is tokenized if followed by a subjunctive. There is an easy fix to this redundancy if you notice, but why run the risk of singleton particle–verb forms, tags, tokens, etc.?

```
|> "s l k" <| [
    FaCaL                      `verb`   [ "proceed", "behave" ]
        `imperf`  FCuL,
    FiCL                       `noun`   [ "wire", "thread" ]
        `plural`  HaFCAL,
    FiCL |< Iy                 `adj`    [ "wire", "by wire" ],
    lA >| FiCL |< Iy           `adj`    [ "wireless", "radio" ],
    FuCUL                      `noun`   [ "behavior", "conduct" ],
    FuCUL |< Iy                `adj`    [ "behavioral" ],
    MaFCaL                     `noun`   [ "road", "method" ]
        `plural`  MaFACiL                                       ]
```

proceed, behave	I(*u*) *salak* سَلَك
wire, thread	(*ʾaslāk* أَسلَاك) *silk* سِلك
wire, by wire	*silkīy* سِلكِيّ
wireless, radio	*lā-silkīy* لَاسِلكِيّ
behavior, conduct	*sulūk* سُلُوك
behavioral	*sulūkīy* سُلُوكِيّ
road, method	*maslak* مَسلَك
	(*masāliku* مَسَالِكُ)

Figure 1: Excerpt from the ElixirFM lexicon of the entries nested under the *s l k* سلك root, and a layout generated from it.

Form	sayad~aEiy	سَيَدَّعِي
Morph	sa + ya + d~aEiy + (null)	
Tag	FUT+IV3MS+IV+IVSUFF_MOOD:I	
Gloss	will + he / it + allege / claim / testify + [ind.]	
Lemma	[Aid~aEaY_1]	اِدَّعَى

With PADT, the word's description hopes to be more intuitive and explicit, and yet to better explain the non-trivial underlying morphological process:

Form	yadda ʿI	*yaddaʿī* يَدَّعِي	sa	*sa* سَ
Morph	"ya" >>| FtaCI |<< "u"		"sa"	
Tag	VIIA-3MS--		F---------	
Form	idda ʿY	*iddaʿā* اِدَّعَى	sa	*sa* سَ
Morph	IFtaCY		"sa"	
Root	"d ʿ w"		"s"	
Reflex	allege, claim, testify		*future marker*	
Class	verb		particle	

Irrespective of the weak character of the morphophonemic pattern, the suffixation of |<< "u" is common to all third person singular indicative imperfective verbs, plus others. Similarly for the subjunctive and jussive templates:

"ya" >>| FtaCI |<< "a"	yadda ʿiya	*yaddaʿiya* يَدَّعِيَ
"ya" >>| FtaCI |<< ""	yadda ʿi	*yaddaʿi* يَدَّعِ

Compare that with the prototypical regular conjugation:

"ya" >>| FCuL |<< "u"	yaktubu	*yaktubu* يَكْتُبُ
"ya" >>| FCuL |<< "a"	yaktuba	*yaktuba* يَكْتُبَ
"ya" >>| FCuL |<< ""	yaktub	*yaktub* يَكْتُبْ

Unlike the Buckwalter analyzer, ElixirFM is suited for both morphological analysis and generation, and can be used as an advanced multi-purpose morphological model. In the interactive mode, one can invoke various utility functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings, exporting and pretty-printing of the information, etc., as well as explore the definitions of the underlying linguistic rules and data being involved. The ElixirFM source code and the lexicon itself are highly reusable by both computers and humans, cf. Figure 1.

Morphological disambiguation of Arabic encompasses subproblems like tokenization, 'part-of-speech' and full morphological tagging, lemmatization, diacritization, restoration of the structural components of words, and combinations thereof. Given a list of possible readings of an input string produced by an analyzer, it can be worthwhile to organize the analyses into a MorphoTrees hierarchy (Smrž and Pajas, 2004) with the string as its root and the full tokens as the leaves, grouped by their lemmas, non-vocalized canonical forms, and partitionings of the string into such forms. The shift from lists to trees enables clearer presentation of the options and their more convenient annotation. MorphoTrees promote gradual focusing on the solution through efficient top-down search or bottom-up pruning using restrictions on the properties of the nodes, and allow inheritance, refinement and sharing of information.

MorphoTrees in Figure 2 depict a complex annotation of the string fhm فهم resolved as *fa-hum* 'so they'. Alternative ways of annotating and details on the automation of some of the steps in the process are explained in (Smrž, 2007b).

3. Dependency Syntax

Morphological annotations identify the textual forms of a discourse lexically and recognize their grammatical properties. The analytical syntactic processing describes the superficial dependency structures in the discourse, whereas the tectogrammatical representation reveals the underlying dependency structures and restores the linguistic meaning.

Annotations on the analytical level are represented by dependency trees. Their nodes map, one to one, to the tokens resulting from the morphological analysis and tokenization, and their roots group the nodes according to the division into sentences or paragraphs. Edges in the trees show that there is a syntactic relation between the governor and its dependent, or rather, the whole subtree under and including the dependent. The nature of the government is expressed by the analytical functions of the nodes being linked.

Figures 3 and 4 analyze the following sentences:

> . . . بعد أن أمضى فيها نحو عشرين عاما.
>
> '. . . after he had spent in it almost twenty years.'
>
> في ملف الأدب طرحت المجلة قضية اللغة العربية
> والأخطار التي تهددها.
>
> 'In the section on literature, the magazine presented the issue of the Arabic language and the dangers that threaten it.'

The connection between the PADT dependency analytical trees and the phrase-structure trees of PATB was studied in

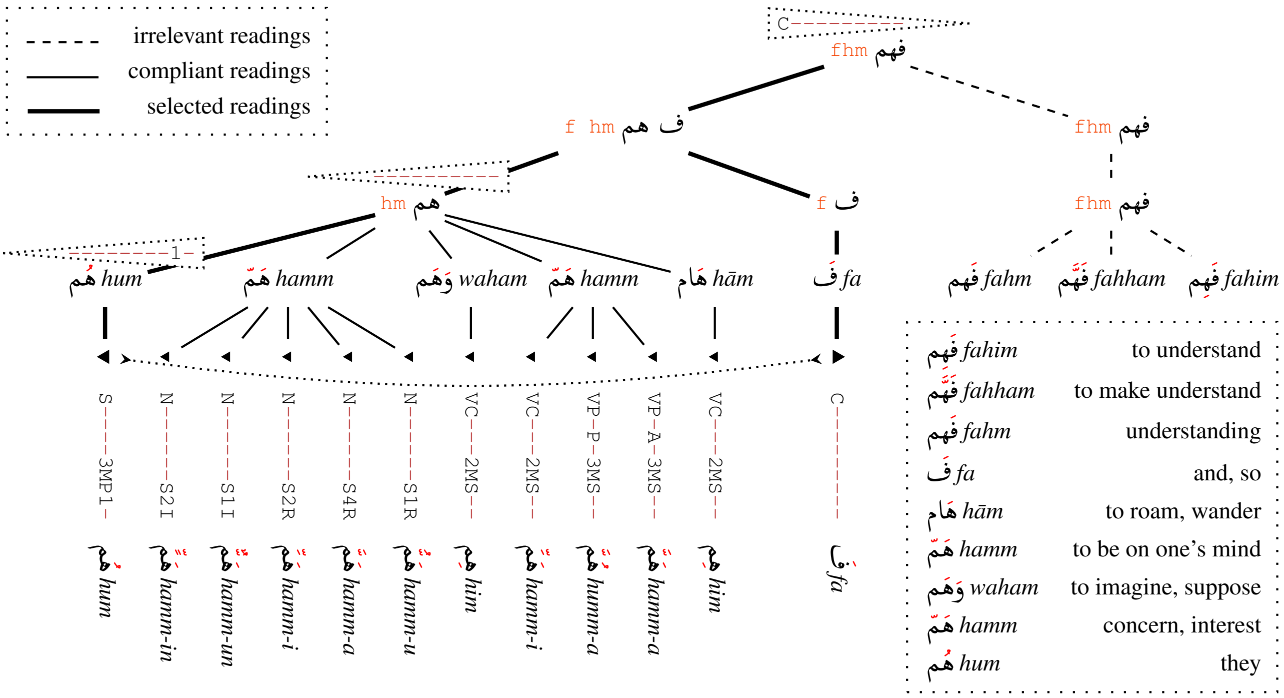
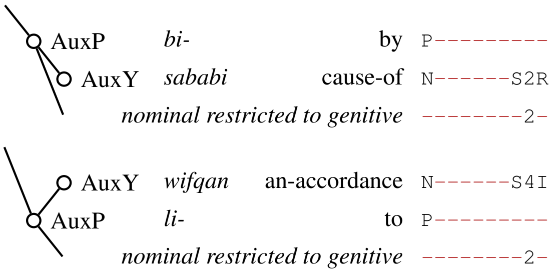


Figure 2: MorphoTrees of the orthographic string fhm فهم including annotation with restrictions. The dashed lines indicate there is no solution suiting the inherited restrictions in the given subtree. The dotted arc symbolizes that there can be implicit morphosyntactic constraints between the adjacent tokens in the analyses, the consistency of which should be verified.

(Žabokrtský and Smrž, 2003). Other notable insights are given in (Habash and Rambow, 2004; Habash et al., 2007). The conversion from phrase-structure trees to dependencies needs to translate the topology of the original data and assign new labels to the nodes of the resulting trees. In some implementations, these tasks are strictly decoupled, and the translation rests in appointing the head subtree out of the children of a particular type of phrase. As illustrated below, this pure percolation mechanism attaching non-head subtrees to the head cannot account for all desired reconfigurations, requiring yet another kind of a translation procedure to be developed and applied on the temporary result. Our implementation of the conversion consists of more consecutive phases as well, but it attempts to use a stronger and more uniform formalism in both the ConDep primary phase and the DepDep secondary phase. Individual conversion rules specify subsequences of nodes and a subroutine that should be triggered if their pattern appears in the data. A rule’s subroutine can manipulate the matching nodes rather freely and can return more than one transformed subtree to be integrated into the larger result, allowing more diverse attachments than what pure percolation achieves.

In Figure 3, we present a rule that transforms a SBAR containing a preposition *baʿda*, a conjunction *ʿan*, and some unspecified other children, in our case the adverbial S clause. This rule would also match on any uninflected preposition followed by *ʾanna*. The subroutine calls the recursive ConDep procedure on all the children, attaches the conjunction to the preposition and the rest of the nodes to the conjunction, and assigns the analytical labels AuxP and AuxC to the respective nodes. The preposition with its subtree is then returned. Other SBAR rules would replace the . . . .

One of the differences between PATB and PADT that must be taken into account when converting the syntactic data is the formal treatment of Arabic compound prepositions, such as *bi-sababi* ‘because of’, *wifqan li-* ‘according to’, *bi-’n-nisbati ʾilā* ‘with respect to’. While PATB does not explicitly distinguish this phenomenon, the PADT approach inspired by the dependency formalism of the Prague Dependency Treebank for Czech does provide a formal solution to these and similar cases (Hajič et al., 1999: 162–163). The criteria for regarding multi-word expressions as compound prepositions are that they be well established (usually listed in dictionaries) and not make sense when standing alone without a following nominal phrase. Compound prepositions were gradually gathered during annotations of the PADT analytical syntax, and can as well be extracted from the data. Their lists have been used in the conversion procedure of PATB and in annotation consistency checks. The reattachments of the nodes in the compounds are implemented in the DepDep dependency tree ‘parser’, which matches on nodes in their linear order, but retains, accesses, and possibly modifies their dependency information.


AuxP bi- by P---------
AuxY sababi cause-of N------S2R
nominal restricted to genitive --------2-
AuxY wifqan an-accordance N------S4I
AuxP li- to P---------
nominal restricted to genitive --------2-

If a certain compound preposition consists of a preposition and a noun regardless of their word order, e.g. *bi-sababi* or *wifqan li-*, the noun as well as the following complement always depend on the preposition, which becomes the head

```
'SBAR'  =>  [ [

  [ ["P-------4-", "ba`da|qabla"],
    ["C---------", "'an"],
    undef ],

  [ ["P---------", ".+"],
    ["C---------", "'anna"],
    undef ],

  sub { my ($root, undef, @data) = @_;

    my @node = map { ConDep($_) } @data;

    PasteNode($node[1], $node[0]);

    PasteNode($_, $node[1]) foreach
                 @node[2 .. @node - 1];

    $node[0]->{'afun'} = "AuxP";
    $node[1]->{'afun'} = "AuxC";

    return $node[0] }

  ], ... ]
```

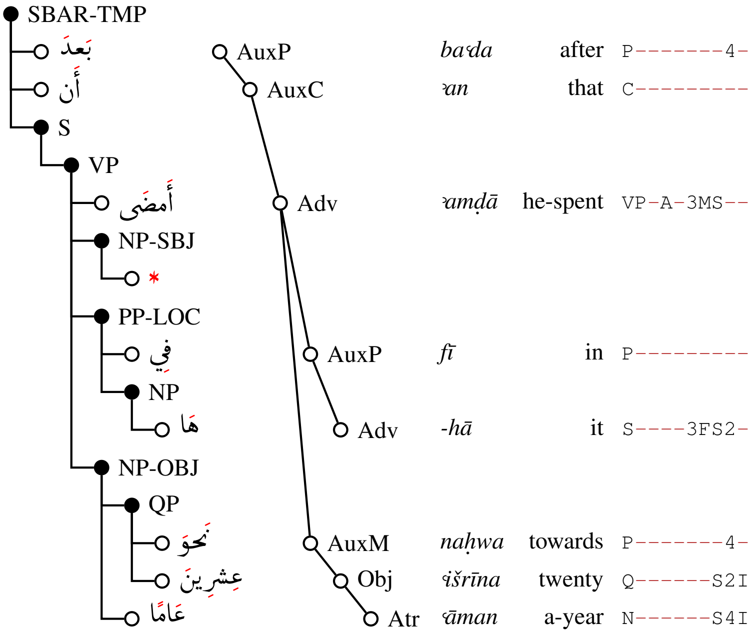
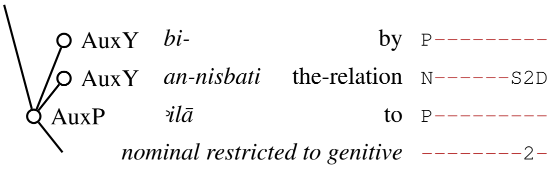
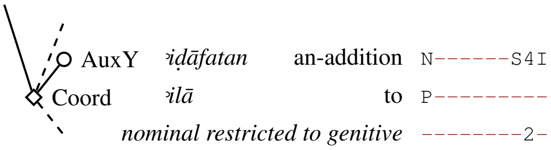

SBAR-TMP
بَعدَ
أَن
S
VP
أَمضَى
NP-SBJ
*
PP-LOC
فِي
NP
هَا
NP-OBJ
QP
نَحوَ
عِشرِينَ
عَامًا
AuxP baʿda after P-------4-
AuxC ʾan that C---------
Adv ʾamḍā he-spent VP-A-3MS--
AuxP fī in P---------
Adv -hā it S----3FS2-
AuxM naḥwa towards P-------4-
Obj ʿišrīna twenty Q------S2I
Atr ʿāman a-year N------S4I

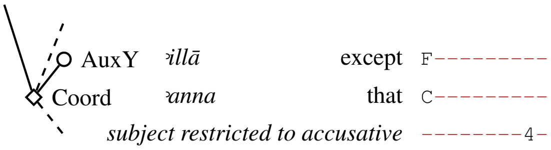
Figure 3: ConDep conversion rule applied to the top level of the phrase-structure tree, and the resulting dependency tree.

AuxP of that prepositional phrase. The nominal part of the compound preposition bears the auxiliary function AuxY, whereas the complement receives its analytical function according to the role of the whole prepositional phrase in a sentence. If a compound preposition is composed of a noun and two prepositions, e.g. *bi-'n-nisbati ʾilā*, the last preposition in the string becomes the head AuxP, and the remaining components depend on it side by side bearing the auxiliary functions AuxY. The complement that follows this compound preposition also hangs on the head AuxP and bears the analytical function determined by the context.

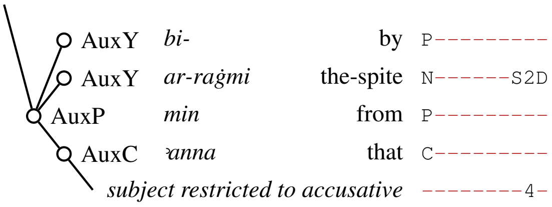

AuxY bi- by P---------
AuxY an-nisbati the-relation N------S2D
AuxP ʾilā to P---------
nominal restricted to genitive --------2-

The motivation for this explicit identification of compound prepositions is that on the tectogrammatical layer, these compounds as well as all other synsemantic words disappear. Only the nodes of autosemantic words remain, representing their underlying syntactic and semantic roles.


AuxY ʾiḍāfatan an-addition N------S4I
Coord ʾilā to P---------
nominal restricted to genitive --------2-


AuxY ʾillā except F---------
Coord ʾanna that C---------
subject restricted to accusative --------4-

Very similar to compound prepositions is our formal treatment of compound conjunctions, e.g. *ʾillā ʾanna* 'however', *ʾiḍāfatan ʾilā* 'in addition to'. In these cases, one of the components is appointed the head (Coord in coordination, AuxC otherwise), while the other one attaches to it as AuxY. Multi-word expressions like *baʿda ʾan* 'after', *qabla ʾan* 'before', *bi-'r-raġmi min ʾanna* 'in spite of the fact that, although' are not considered to be compound conjunctions. They are regarded as prepositions or compound prepositions followed by a conjunction, due to the fact that the conjunction and its clause can in general be replaced by a nominal phrase.

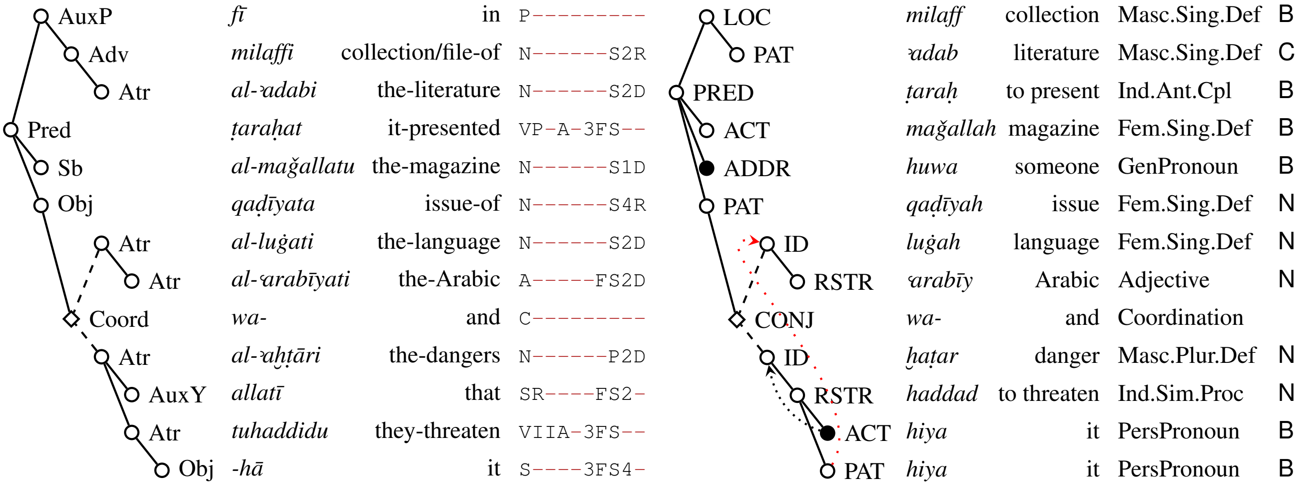

AuxY bi- by P---------
AuxY ar-raġmi the-spite N------S2D
AuxP min from P---------
AuxC ʾanna that C---------
subject restricted to accusative --------4-

There are many other kinds of syntactic differences between PATB and PADT. Some structures in PATB tend to be more semantically oriented, while others are rather simplified. Note e.g. the use of QP in Figure 3, which breaks the grammatical dependency between the modifier *naḥwa*, the numeral, and the counted object (cf. Habash et al., 2007). The favored PATB annotation would, however, correspond to the tectogrammatical treatment of quantifiers in PADT.

Both ConDep and DepDep tackle even issues due to certain inconsistency of annotations. Flat phrases must be parsed, improper dependencies eliminated, incorrect instances of subordination need to be restructured to coordination, etc.

The functional labels in PATB capture several types of adverbials, conflated to Adv on the analytical level. On the other hand, the set of tectogrammatical functors in PADT is yet much more refined (cf. Mikulová et al., 2006).

The complete tree conversion process includes also the resolution of traces. This phase yields pointers of grammatical coreference, present in the manual analytical data as well (Hajič et al., 2004b). The recovery of textual coreference is performed on the tectogrammatical level, cf. Figure 4.



AuxP	*fī*	in	`P---------`
Adv	*milaffi*	collection/file-of	`N------S2R`
Atr	*al-ʾadabi*	the-literature	`N------S2D`
Pred	*ṭaraḥat*	it-presented	`VP-A-3FS--`
Sb	*al-mağallatu*	the-magazine	`N------S1D`
Obj	*qaḍīyata*	issue-of	`N------S4R`
Atr	*al-luġati*	the-language	`N------S2D`
Atr	*al-ʿarabīyati*	the-Arabic	`A-----FS2D`
Coord	*wa-*	and	`C---------`
Atr	*al-ʾaḫṭāri*	the-dangers	`N------P2D`
AuxY	*allatī*	that	`SR----FS2-`
Atr	*tuhaddidu*	they-threaten	`VIIA-3FS--`
Obj	*-hā*	it	`S----3FS4-`

LOC	*milaff*	collection	Masc.Sing.Def	B
PAT	*ʾadab*	literature	Masc.Sing.Def	C
PRED	*ṭaraḥ*	to present	Ind.Ant.Cpl	B
ACT	*mağallah*	magazine	Fem.Sing.Def	B
ADDR	*huwa*	someone	GenPronoun	B
PAT	*qaḍīyah*	issue	Fem.Sing.Def	N
ID	*luġah*	language	Fem.Sing.Def	N
RSTR	*ʿarabīy*	Arabic	Adjective	N
CONJ	*wa-*	and	Coordination	
ID	*ḫaṭar*	danger	Masc.Plur.Def	N
RSTR	*haddad*	to threaten	Ind.Sim.Proc	N
ACT	*hiya*	it	PersPronoun	B
PAT	*hiya*	it	PersPronoun	B

Figure 4: *Left:* Example of analytical annotation. Orthographic words are tokenized into lexical words, and their inflectional morphosyntactic properties are encoded using the positional tags. Coordination members are depicted with dashed edges. *Right:* Example of tectogrammatical annotation with resolved coreference (extra arcs) and indicated values of contextual boundness. Lexemes are identified by lemmas, and selected grammatemes are shown in place of morphosyntactic features.

4. Tectogrammatics

Tectogrammatics, the underlying syntax reflecting the linguistic meaning of an utterance, is the highest level of annotation in the family of Prague Dependency Treebanks (Hajič et al., 2006). It captures dependency and valency (Žabokrtský, 2005) with respect to the deep linguistic relations of discourse participants. In its generality, the description also includes topic–focus articulation, coreference resolution, and other non-dependency relations. The set of tectogrammatical annotations in PADT is still rather experimental, yet, we intend to develop this formalism further.

The topology of a tectogrammatical representation of a sentence is similar to that of the analytical level. In contrast to it, nodes in the tree may be deleted, inserted, and even reorganized. We speak of a transfer of structures from analytical to tectogrammatical, which can be partly automated. The nodes appear as lexical entries rather than inflected forms. Grammatemes, the deep grammatical parameters, abstract away from the morphological and analytical features of an utterance. Functors, the deep roles that the participants assume, include Actor, Patient, Addressee, Origin, Effect, various types of local and temporal modifications, Extent, Manner, Cause, Identity, Restriction, coordination types, and many more (Mikulová et al., 2006).

Figure 4 compares the analytical and tectogrammatical representations of a sentence. The inserted nodes are recovered from the discourse as the obligatory actants of the valency frames of the two verbs (cf. Bielický and Smrž, 2008). Values of contextual boundness, a feature from which the topic–focus dichotomy is inferred, are also indicated (cf. Hajičová and Sgall, 2003; Smrž and Hajič, 2008).

There is a number of structures on the analytical level that appear as a co-sign, i.e. their actual meaning results only from being combined. Quite often, one of the nodes in a structure is occupied by a verb while its other members are modifiers, which are not further developed.

The tectogrammatical level (t-level) erases some language-specific features present on the analytical level. The treatment of Arabic verbal negation is an instance thereof. Generally, verbal negation on the t-level is expressed by an abstract node for negation. The reason for it being a node and not a grammateme is that the deep ordering of the node with respect to the verb determines the scope of negation, with consequences for the information structure of a discourse.

In Arabic, a variety of combinations, such as *lam yaktub* vs. *mā kataba* ‘he did not write’, turn into exactly the same structures on the t-level. Stylistic variation is not reflected in tectogrammatics—it is believed that *lam yaktub* is more formal, while *mā kataba* is considered rather dialectal or used only in spoken discourse. The *mā* type of negation is used in rather fixed collocations, such as *mā zāla* ‘still’.

The opposition of perfective *katab* and imperfective *yaktub* forms in Arabic, generally perceived as an opposition of past and non-past, is actually irrelevant for the deep notion of tense in this context. On the analytical level, the tense reference is expressed as a co-sign consisting of a negative particle and a finite verbal form. On the t-level, the tense indication is marked only in the grammatemes of the verbal node, i.e. [tense=Ant] for the anterior tense.

The other markers of verbal negation, *lā* and *lan*, are represented in the same manner, but relate to different tenses.

Structurally very similar on the analytical level is the use of a future marker for expressing positive posterior tense. In *sa-yaktubu* ‘he will write’, the modifier *sa-* is attached to the node of the verb, and itself has no offsprings. The more explicit form *sawfa* can also be used. In combination of *sawfa* with negation, the particle *lā* is inserted between the marker and the verb, i.e. *sawfa lā yaktubu*. However, *sa-* cannot combine with negation, in which case *lan* is used instead, as in *lan yaktuba* ‘he will not write’. On the t-level, the modifier node containing the future marker is always deleted and the tense is indicated at the verb [tense=Post].

An interesting point is the treatment of *qad*, a modifying particle attached to verbs. When connected with the perfective form of a verb, it has the meaning of an aspectual nuance of completed action, like ‘already’. On the t-level, this particle is deleted and the verbal node receives the grammatemes for anteriority and completeness [tense=Ant, aspect=Cpl]. However, when used with an imperfective verbal form, its meaning changes to possibility in the future, ‘it might well be that’. The grammatemes of the verb become [tense=Post, deontmod=Poss], but the modifier node is retained on the t-level, as is the case with other kinds of modality nodes (cf. Mikulová et al., 2006).

5. Quality Control

Our software environment for maintaining the PADT and PATB data is TrEd, an editor for tree-like structures developed in Perl by Petr Pajas. It is a highly customizable and programmable tool providing both the graphical user interface and the application programming interface used for network-oriented data processing, such as conversions or consistency checks (cf. Štěpánek, 2006).

TrEd integrates all the levels of annotation by enabling the user to invoke macros or external programs of any kind. During annotation, one can take great advantage of specific contexts/modes with predefined macro operations, keyboard-shortcuts, and stylesheets for informative display of the data. The dependency approach to syntax does not restrict TrEd itself. Next to MorphoTrees (Smrž and Pajas, 2004), we have for instance implemented contexts for viewing and possibly annotating phrase-structure trees.

Most of our quality control programs are written as TrEd scripts. Linguistic and formal constraints on the annotated data, as well as requirements on the mutual compatibility between levels, are implemented with transparent code reusable also in automatic tagging and partial parsing.

Using this technology, we have successfully discovered and eliminated many annotation errors and inconsistencies in both PADT and PATB, on all the levels of annotation. We will report on this process in detail in the documentation to PADT 2.0. The order of revisions is in tens of thousands of words, i.e. percents of the whole treebank, which means great improvement in the accuracy of PADT 2.0, and might have significant effect on the performance of any derived applications (cf. Habash et al., 2007).

Our experiments on an earlier version of the data indicated that annotating MorphoTrees is up to three times faster than disambiguating morphology in form of the classical MorphoLists. The inter-annotator disagreement on MorphoTrees was 5.3 %, on MorphoLists it reached 9.3 %. The inter-annotator disagreement in the attachment of nodes on the analytical level before revisions was measured to 9.2 %. With the upgrade of the morphological data in the whole treebank to ElixirFM, as well as with the introduction of our new tools for consistency verification and data processing, we believe that the inter-annotator agreement will yet considerably improve on these values.

6. Conclusion

In this contribution, we have overviewed the theoretical concepts behind the Prague Arabic Dependency Treebank, and have discussed converting the Penn Arabic Treebank into the PADT style. We have described the original data and the tools that we develop. PADT 2.0 will be an important new linguistic resource. TrEd with its extensions, ElixirFM, and Encode Arabic are open-source software published under the GNU General Public License:

```
http://ufal.mff.cuni.cz/~pajas/tred/
http://sf.net/projects/elixir-fm/
http://sf.net/projects/encode-arabic/
```

Acknowledgements

We would like to thank all our colleagues in the PADT project for their annotation efforts, inspiring ideas and valuable suggestions, and for their long-time support: Tereza Čečáková, Milada Frantová, Petr Pajas, Pavel Pecina, Zdeněk Žabokrtský, and others.

This research has been funded by the Ministry of Education of the Czech Republic (MSM0021620838 and MSM0021-620823) and by the Grant Agency of the Czech Academy of Sciences (1ET101120413).

References

Viktor Bielický and Otakar Smrž. Building the Valency Lexicon of Arabic Verbs. In *LREC 2008: Sixth Language Resources and Evaluation Conference*, Marrakech, Morocco, 2008.

Ann Bies. English-Arabic Treebank v 1.0. LDC2006T10, ISBN 1-58563-387-9, 2006.

Tim Buckwalter. Issues in Arabic Orthography and Morphology Analysis. In *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages 31–34, Geneva, 2004a.

Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 1.0. LDC2002L49, ISBN 1-58563-257-0, 2002.

Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 2.0. LDC2004L02, ISBN 1-58563-324-0, 2004b.

Simon Corston-Oliver, Anthony Aue, Kevin Duh, and Eric Ringger. Multilingual Dependency Parsing using Bayes Point Machines. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 160–167, Morristown, NJ, 2006.

Wolfdietrich Fischer. *A Grammar of Classical Arabic*. Yale Language Series. Yale University Press, third revised edition, 2002. Translated by Jonathan Rodgers.

Markus Forsberg and Aarne Ranta. Functional Morphology. In *Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming, ICFP 2004*, pages 213–223. ACM Press, 2004.

David Graff. Arabic Gigaword Third Edition. LDC2007T40, 1-58563-460-3, 2007.

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, 2005.

Nizar Habash and Owen Rambow. Extracting a Tree Adjoining Grammar from the Penn Arabic Treebank. In *JEP-TALN 2004, Session Traitement Automatique de l'Arabe*, Fes, Morocco, April 2004 2004.

Nizar Habash and Owen Rambow. Arabic Diacritization through Full Morphological Tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56, Rochester, New York, April 2007. Association for Computational Linguistics.

Nizar Habash, Ryan Gabbard, Owen Rambow, Seth Kulick, and Mitch Marcus. Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1084–1092, 2007.

Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel Beška, Jakub Kráčmar, and Kamila Hassanová. Prague Arabic Dependency Treebank 1.0. LDC2004T23, ISBN 1-58563-319-4, 2004a.

Jan Hajič, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. Prague Arabic Dependency Treebank: Development in Data and Tools. In *NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117. ELDA, 2004b.

Jan Hajič, Otakar Smrž, Tim Buckwalter, and Hubert Jin. Feature-Based Tagger of Approximations of Functional Arabic Morphology. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pages 53–64, Barcelona, Spain, 2005.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, and Marie Mikulová. Prague Dependency Treebank 2.0. LDC2006T01, ISBN 1-58563-370-4, 2006.

Jan Hajič et al. A Manual for Analytic Layer Annotation of the Prague Dependency Treebank. Technical Report 28, UFAL MFF UK, Charles University in Prague, 1999.

Eva Hajičová and Petr Sgall. Dependency Syntax in Functional Generative Description. In *Dependenz und Valenz – Dependency and Valency*, volume I, pages 570–592. Walter de Gruyter, 2003.

Seth Kulick, Ryan Gabbard, and Mitch Marcus. Parsing the Arabic Treebank: Analysis and Improvements. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT 2006)*, pages 31–42, Prague, Czech Republic, 2006.

Xiaoyi Ma. Arabic English Parallel News Part 1. LDC2004T18, ISBN 1-58563-310-0, 2004.

Mohamed Maamouri and Ann Bies. Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools. In *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages 2–9, Geneva, 2004.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Hubert Jin. Arabic Treebank: Part 2 v 2.0. LDC2004T02, ISBN 1-58563-282-1, 2004a.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR International Conference on Arabic Language Resources and Tools*, pages 102–109. ELDA, 2004b.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Hubert Jin. Arabic Treebank: Part 1 v 3.0. LDC2005T02, ISBN 1-58563-330-5, 2005a.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Hubert Jin, and Wigdan Mekki. Arabic Treebank: Part 3 v 2.0. LDC2005T20, ISBN 1-58563-341-0, 2005b.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Hubert Jin, and Wigdan Mekki. Arabic Treebank: Part 4 v 1.0. LDC2005T30, ISBN 1-58563-343-7, 2005c.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.

Marie Mikulová et al. A Manual for Tectogrammatical Layer Annotation of the Prague Dependency Treebank. Technical Report 30, UFAL MFF UK, Charles University in Prague, 2006.

Rani Nelken and Stuart M. Shieber. Arabic Diacritization Using Finite-State Transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 79–86, Ann Arbor, 2005.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, 2007.

Karin C. Ryding. *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press, 2005.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel & Academia, 1986.

Noah A. Smith, David A. Smith, and Roy W. Tromble. Context-Based Morphological Disambiguation with Random Fields. In *Proceedings of HLT/EMNLP 2005*, pages 475–482, Vancouver, 2005. Association for Computational Linguistics.

Otakar Smrž. ElixirFM — Implementation of Functional Arabic Morphology. In *ACL 2007 Proceedings of the Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 1–8, Prague, Czech Republic, June 2007a. Association for Computational Linguistics.

Otakar Smrž. *Functional Arabic Morphology. Formal System and Implementation*. PhD thesis, Charles University in Prague, 2007b.

Otakar Smrž and Jan Hajič. The Other Arabic Treebank: Prague Dependencies and Functions. In Ali Farghaly, editor, *Arabic Computational Linguistics: Current Implementations*. CSLI Publications, 2008.

Otakar Smrž and Petr Pajas. MorphoTrees of Arabic and Their Annotation in the TrEd Environment. In *NEMLAR International Conference on Arabic Language Resources and Tools*, pages 38–41. ELDA, 2004.

Jan Štěpánek. Post-annotation Checking of Prague Dependency Treebank 2.0 Data. In *Proceedings of the 9th International Conference TSD 2006*, number 4188 in Lecture Notes in Computer Science, pages 277–284. Springer-Verlag, 2006.

Gregory T. Stump. *Inflectional Morphology. A Theory of Paradigm Structure*. Cambridge Studies in Linguistics. Cambridge University Press, 2001.

Jim Yaghi and Sane Yagi. Systematic Verb Stem Generation for Arabic. In *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages 23–30, Geneva, 2004.

Zdeněk Žabokrtský. *Valency Lexicon of Czech Verbs*. PhD thesis, Charles University in Prague, 2005.

Zdeněk Žabokrtský and Otakar Smrž. Arabic Syntactic Trees: from Constituency to Dependency. In *EACL 2003 Conference Companion*, pages 183–186, Budapest, Hungary, 2003.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia, July 2006. Association for Computational Linguistics.

Arabic Named Entity Recognition using Conditional Random Fields

Yassine Benajiba and Paolo Rosso

Natural Language Engineering Lab.
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera, s/n, 46022, Valencia (España)
{ybenajiba,prosso}@dsic.upv.es

Abstract

The Named Entity Recognition (NER) task consists in determining and classifying proper names within an open-domain text. This Natural Language Processing task proved to be harder for languages with a complex morphology such as the Arabic language. NER was also proved to help Natural Language Processing tasks such as Machine Translation, Information Retrieval and Question Answering to obtain a higher performance. In our previous works we have presented the first and the second version of ANERsys: an Arabic Named Entity Recognition system, whose performance we have succeeded to improve by more than 10 points, from the first to the second version, by adopting a different architecture and using additional information such as Part-Of-Speech tags and Base Phrase Chunks. In this paper, we present a further attempt to enhance the accuracy of ANERsys by changing the probabilistic model from Maximum Entropy to Conditional Random Fields which helped to improve the results significantly.

1. Introduction

The Named Entity Recognition (NER) task consists in determining and classifying proper names in an open-domain text. Many research works have been conducted to prove the predominant importance of NER to the other Natural Language Processing (NLP) tasks; some of these investigations are the following:

- In *Machine Translation* (MT), NEs require different techniques of translation than the rest of words of the text. Also, the post-editing step is more expensive when the errors of a MT system are mainly in NEs translation. For these reasons, (Babych and Hartley, 2003) have carried out a research study where they tag a text with a NER system as a pre-processing step of MT. The authors report that they have reached a higher accuracy with this new approach which helps the MT system to switch to a different translation technique when a Named Entity (NE) is detected.

- *Search results clustering*, is a sub-task of text clustering. It consists of organizing in groups the results returned by an IR system in order to make them easier to read for the user. In (Toda and Kataoka, 2005), the authors argue that they outperform the existing search results clustering techniques by including a NER system in their global system in order to give a special weight to the NEs in their clustering approach.

- *Information Retrieval* (IR), is a task which aims at retrieving the relevant document for the query formulated by the user in natural language: (Thompson and Dozier, 1997) report that 67.83%, 83.4% and 38.8% of the queries contained one or more Named Entities (NEs) according to Wall St. Journal, Los Angeles Times and Washington Post, respectively. Hence, an improvement of the retrieval of documents for queries which contain NEs would boost significantly the performance of the global IR system. In their research study, the authors have explored an approach which treats NEs and non-NEs differently. Their results show that the IR system precision outperforms the results obtained by a probabilistic retrieval engine on all the recall levels.

- *Question Answering* (QA), one of the most complicated NLP tasks because at satisfying the need of a special type of users which ask for an accurate answer to a specific question. Thus, a QA system does not stop at retreiving the relevant documents (like an IR system), it has also to answer but it has also to automatically extract the answer. In order to do so, a QA system has to perform several steps of processing both the question and the document-set where the system retrieves the answer (Benajiba et al., 2007). Many are the studies which show that the accuracy of a QA system relies significantly on the performance of the NER system included within, such as: (Ferrandez et al., 2007) which explore the accuracy of the global, both monolingual and cross-lingual, QA system for different NER systems. (Greenwood and Gaizauskas, 2007) use a NER system in order to improve the performance of an answer extraction module based on a pattern-matching approach. The authors use the NER system to capture the answers which are not possible to capture using only patterns. They report improving that the accuracy of answering the questions of type “When did X die” from 0% to 53%. (Mollá et al., 2006) also conducted a research study of the improvement obtained when the NER system tag-set corresponds exactly to the classes of NEs retrieved by the

QA system. The final results showed that up to 1.3% of improvement can be obtained in case both the NER system and global QA system aim at the same classes of NEs.

In order to use a standard definition of the NER task we have used the definition which was formulated in the in the shared task of the Conferences on Computational Natural Language Learning (CoNLL). In the sixth and the seventh editions of the Conference on Computational Natural Language Learning (CoNLL 2002[1] and CoNLL 2003[2]) the NER task was defined as to determine the proper names existing within an open domain text and classify them as one of the following four classes:

1. *Person*: named person or family;
2. *Location*: name of politically or geographically defined location;
3. *Organization*: named corporate, governmental, or other organizational entity; and
4. *Miscellaneous*: the rest of proper names (vehicles, weapons, etc.).

In the literature, very few research works were oriented especially to the NER task for Arabic texts (Abuleil, 2002; Maloney and Niv, 1998). Moreover, most of the effort were done for commercial purposes: Siraj[3] (by Sakhr), ClearTags[4] (by ClearForest), NetOwlExtractor[5] (by NetOwl) and InxightSmartDiscoveryEntityExtractor[6] (by Inxight). Unfortunately, no performance accuracy nor technical details have been provided and a comparative study of the systems is not possible. However, during the two editions of the CoNLL which we have previously mentioned, many research works addressed the language-independent NER task. A general study of these works showed that Maximum Entropy is an efficient approach for the task in question (Bender et al., 2003; Chieu and Ng, 2003; Curran and Clark, 2003; Cucerzan and Yarowsky, 1999; Malouf, 2003).

Recently, the Conditional Random Fields (CRF) model (Lafferty et al., 2001) proved to be very successful in many NLP tasks such as: shallow parsing (Sha and Pereira, 2003), morphological analysis (Kudo et al., 2004), information extraction (Pinto et al., 2003), biomedical NER (Settles, 2004), etc. Moreover, CRF proved a special success in the NER task for many languages of different levels of morphological complexity:

(i) *English* and *German*: (McCallum and Li, 2003) is one of the first attempts of using CRF for the NER task. The authors used the CoNLL 2003 corpus for evaluation and they report in their paper that an accuracy (F-measure) of 68.11 was reached for German, whereas 84.04 was obtained for English;

(ii) *Vietnamese*: in (Tran et al., 2007) a comparative study of Support Vector Machine (SVM) vs. CRF has been done and the results showed that using CRFs they have reached an accuracy of 86.48 vs. 87.75 using SVM. However, the authors report various experiments using different context window sizes for the SVM approach evaluation, whereas just one single result is reported for the CRF approach;

(iii) *Hindi*: 71.5 was reached for this language in (Li and McCallum, 2003) using CRF. However, the authors report that they have used a feature-induction technique because of their ignorance of the Hindi language peculiarities; and

(iv) *Chinese*: (Wu et al., 2006) reports in the paper that they have two different corpora for evaluation. For the first corpus, the best results were obtained when they used a combination of CRF and Maximum Entropy, whereas for the second corpus the best results were obtained for CRF. Moreover, the authors report that the worst results have been obtained when they have combined different CRF models.

To our knowledge, up to now there is no research study which has been carried out in order to prove the efficiency of the CRF model for NER in Arabic texts. Therefore, the idea behind the research work we present in this paper is to conduct experiments to investigate the performance of the CRF model for the Arabic NER task taking into consideration the peculiarities of the Arabic language and comparing the obtained results with our previous experiments which have been conducted using a Maximum Entropy approach.

The rest of this paper is structured as follows. In the second section of this paper we will give an overview of the Arabic language peculiarities. Section Three will describe our previous works related to the NER task. Section Four is dedicated to give a brief description of the CRF model. Details about the evaluation data we use in our experiments are given in Section Five. Finally, in the sixth section we present the results of our preliminary experiments with CRF and a comparison with our previous works results, whereas in the seventh section we draw some conclusions and discuss future works.

2. The Challenges of Arabic Named Entity Recognition

From a general viewpoint, the NER task can be considered as a composition of two sub-tasks:

1. *The detection of the existing NEs in a text* Which is a quite easy sub-task if we can use the capital letters as indicators to determine where the NEs start and where they end. However, this is only possible when the capital letters are supported in the target language, which is not the case for the Arabic language (Figure 1 shows the example of two words where only one of them is a NE and both of them start with the same character). The absence of capital letters in the Arabic language is the main obstacle to obtain high performance in NER (Benajiba et al., 2007)(Benajiba and Rosso, 2007).

2. *The classification of the NEs*

[1] http://www.cnts.ua.ac.be/conll2002/

[2] http://www.cnts.ua.ac.be/conll2003/

[3] http://siraj.sakhr.com/

[4] http://www.clearforest.com/index.asp

[5] http://www.netowl.com/products/extractor.html

[6] http://www.inxight.com/products/smartdiscovery/ee/index.php

(mouth)
فم
(Valencia)
فالنسيا

Figure 1: An example illustrating the absence of capital letters in Arabic

The Arabic language is a highly inflectional language, i.e., an Arabic word can be seen as the following composition:

Word = prefix(es) + lemma + suffix(es)

The *prefixes* can be articles, prepositions or conjunctions, whereas the *suffixes* are generally objects or personal/possessive anaphora. Both prefixes and suffixes are allowed to be combinations, and thus a word can have zero or more affixes. From a statistical viewpoint, this inflectional charactersitic of the Arabic language makes Arabic texts, compared to texts written in other languages which have a less complex morphology, more sparse and thus most of the Arabic NLP tasks are harder and more challenging. A full description of how thischaracterstic hardens each of the Arabic NLP goes beyond the scope of this paper. However, concerning the classification sub-task of NER, we can say that: the classification of NEs relies mainly on the word and the context in which it appeared in the text in order to decide the class it belongs to. Moreover, in case of an inflectional language, such as Arabic, both the words and the contexts may appear in different forms and thus a huge training corpus is required in order to obtain a high accuracy.

In order to reduce data sparseness in Arabic texts two solutions are possible:

(i) *Light stemming*: consists of omitting all the prefixes and suffixes which have been added to a lemma to obtain the needed meaning. This solution is convenient for tasks such as Information Retrieval and Question Answering because the prepositions, articles and conjunctions are considered as stop words and are not taken into consideration to decide whether a document is relevant for a query or not. An implementation of this solution was available on Kareem Darwish website[7] which has been unfortunately removed;

(ii) *Word segmentation*: consists of separating the different components of a word by a space character. Therefore, this solution is moreadequate for the NLP tasks which require to keep the different word morphemes such as Word Sense Disambiguation, NER, etc. A tool to perform Arabic word segmentation trained on Arabic Treebank, and obtaining an accuracy of *99.12* for this task, is available on Mona Diab website[8].

In our experiments we have adopted the second solution to reduce sparseness in our data and we draw the obtained results in the sixth section.

3. Our Previous Related Work

We have developed two versions of *ANERsys*, our Arabic NER system. Following we give a brief description of both versions of the system, whereas the results obtained with each of the systems will be given in the sixth section.

3.1. ANERsys 1.0: A Maximum Entropy Approach

As we have mentioned in the introduction of this paper, the Maximum Entropy approach has been very successful in the NER task. This approach is based on a exponential model which can be expressed as:

$$p(c|x) = \frac{1}{Z(x)} * exp(\sum_i \lambda_i . f_i(x,c)) \qquad (1)$$

Z(x) is for normalization and may be expressed as:

$$Z(x) = \sum_{c'} exp(\sum_i \lambda_i . f_i(x,c')) \qquad (2)$$

Where *c* is the class, *x* is a context information and f_i*(x,c)* is the i-th feature.
Maximum Entropy is a very convenient approach for the NER task thanks to its feature-based model. In this version of the system, our feature-set, which is fully *binary*, consisted of:
(i) *Wi*: The concerned word and its class;
(ii) $\{$*Wi-2, Wi-1*$\}$ *and* $\{$*Wi+1, Wi+2*$\}$: The bigrams coming before and after the word, which represent basically the context in which the word appears;
(iii) *Wi exists in a gazetteer*: The use of ANERgazet (see Section Five) as an external resource to enhance the system. The gazetteers were used in a binary way i.e., we have incorporated a binary feature which indicates whether *Wi* is an item of one of our gazetteers or not;
(iv) *Wi-1 is a nationality*: The NEs of class *person*, frequently come after the nationality of the person in question in newspapers articles.

3.2. ANERsys 2.0: A 2-step Approach

The error-analysis of ANERsys 1.0 results showed that the system had difficulties with multi-tokens NEs, i.e., it was harder to detect the Names Entities (NEs) than to classify them. Thus, in the second version of the system we have adopted a 2-step approach which is illustrated in Figure 2. The first step of the system is concerned mainly by detecting the start and the final tokens of each NE, whereas the second step takes care of classifying them (a full description of the system is given in (Benajiba and Rosso, 2007))

4. Conditional Random Fields

CRFs (Lafferty et al., 2001) is a probabilistic framework to segment and label sequence data. It is based on undirected graphical models where the nodes represent the label sequence **y** corresponding to the sequence **x**. CRF model aims at finding the label **y** which maximizes the conditional probability **p(y|x)** for a sequence **x**. The CRF model is

[7] http://www.glue.umd.edu/∼kareem/darwish
[8] http://www1.cs.columbia.edu/∼mdiab/

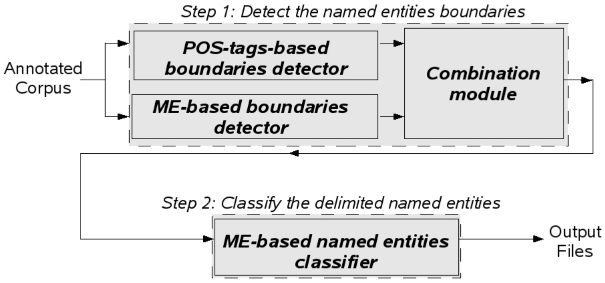

Step 1: Detect the named entities boundaries
Annotated Corpus
POS-tags-based boundaries detector
ME-based boundaries detector
Combination module
Step 2: Classify the delimited named entities
ME-based named entities classifier
Output Files

Figure 2: Generic architecture of ANERsys 2.0

a feature-based model where features have binary values such as:

$f_k(y_{t-1}, y_t, x)$:=1 for x='Darfur' and y_t='B-LOC', and 0 otherwise.

The CRF model is considered a generalization of Maximum Entropy and Hidden Markov Models (HMM) and can be expressed as following:

$$p(y|x) = \frac{1}{Z(x)} * exp(\sum_t \sum_k \lambda_k . f_k(y_{t-1}, y_t, x)) \quad (3)$$

where λ_i represent the weights assigned to the different features in the training phase and **Z(x)** is a normalization factor which can be expressed as:

$$Z(x) = \sum_{y \epsilon Y} exp(\sum_t \sum_k \lambda_k . f_k(y_{t-1}, y_t, x)) \quad (4)$$

5. Evaluation Data

We have used ANERcorp in order to train and test the CRF model. ANERcorp is composed of a training corpus and a test corpus annotated especially for the NER task. We have chosen the tokens of ANERcorp from both news wire and other web resources (more details about ANERcorp are given in (Benajiba et al., 2007)) and we have manually annotated them ourselves. Each token of ANERcorp is tagged as belonging to one of the following classes:

- B-PERS: The Beginning of the name of a PERSon.
- I-PERS: The continuation (Inside) of the name of a PERSon.
- B-LOC: The Beginning of the name of a LOCation.
- I-LOC: The Inside of the name of a LOCation.
- B-ORG: The Beginning of the name of an ORGanization.
- I-ORG: The Inside of the name of an ORGanization.
- B-MISC: The Beginning of the name of an entity which does not belong to any of the previous classes (MISCellaneous).
- I-MISC: The Inside of the name of an entity which does not belong to any of the previous classes.
- O: The word is not a named entity (Other).

ANERcorp contains more than 150,000 tokens (11% of the tokens are part of a NE) and they are freely downloadable from our website[9]. The ANERcorp has been used in our earlier work (Benajiba et al., 2007) (Benajiba and Rosso, 2007) in order to evaluate the two versions of ANERsys which we have described before (see Section Three).

6. Experiments and Results

6.1. Corpus, Baseline, Measure

We have used the ANERcorp (see Section Five) to evaluate our system. The baseline model[10] consists of assigning to a word w_i the class C_i which most frequently was assigned to w_i in the training corpus. The words which were unseen during the training phase are assigned the class *O*. We have used the $F_{\beta=1}$-measure for evaluation:

$$F_{\beta=1} = \frac{(\beta^2 + 1) * precision * recall}{\beta^2 * (precision + recall)} \quad (5)$$

Where *precision* is the percentage of NEs found by the system and which are correct. It can be expressed as:

$$precision = \frac{Num(correct\ NEs\ found)}{Num(NEs\ found)} \quad (6)$$

and *recall* is the percentage of NEs existing in the corpus and which were found by the system. It can be expressed as:

$$recall = \frac{Num(NEs\ found)}{Total\ number\ of\ NEs} \quad (7)$$

6.2. Feature-set

We have kept the same feature-set used in our previous systems (see Section Three) in order to be able to compare the performance of the Maximum Entropy (ME) and the CRF performance.

POS-tag and BPC : The Part-Of-Speech tagging is the task of assiging to each word its linguistic category. Base Phrase Chunks (BPC) are atomic parts of a sentence (beyond words). In CoNLL 2003, the POS-tags, together with the BPC, formed part of the corpora which were provided to the participants (see Figure 3). The point of using POS-tags and BPS relies mainly on that BPC might determine the beginning and the end of a NE and thus help the classifier to capture the boundaries of the NEs. Additionally, using the POS-tags is also helpful thanks to the "*NNP*" tag which marks a word a NE. However, in the proceedings of the conference there were no studies reporting the impact of each of these features individually.

[9] http://www.dsic.upv.es/∼ybenajiba

[10] http://cnts.ua.ac.be/conll2002/ner/bin/baseline

U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

Figure 3: An extract of the CoNLL 2003 English corpus

External Resources (*GAZ*) : In order to measure the impact of using external resources in the NER task we have used ANERgazet (also available on our website) which consists of three different gazetteers, all built manually using web resources:
(i) *Location Gazetteer*: this gazetteer consists of 1,950 names of continents, countries, cities, rivers and mountains found in the Arabic version of wikipedia[11];
(ii) *Person Gazetteer*: this was originally a list of 1,920 complete names of people found in wikipedia and other websites. After splitting the names into first names and last names and omitting the repeated names, the list contains finally 2,309 names;
(iii) *Organizations Gazetteer*: the last gazetteer consists of a list of 262 names of companies, football teams and other organizations.

W_{i-1} **is a Nationality (*NAT*)** : Frequently, NEs of the class "Person" comes after mentioning the nationality of the person (especially in newspaper articles). For instance, *the Iranian Presiden Mahmoud declared ....*

6.3. Results

Baseline and Previous Results Table 1 shows the baseline results. Tables 2 and 3 show the results obtained, respectively, by the first and the second version of ANERsys. Using a ME approach (ANERsys 1.0) has helped to obtain an F-measure which is almost 12 points above the baseline (55.23). Moreover, when we have used a 2-step approach and adopted different techniques for detecting and classifying the NEs, we have significantly raised the recall of our system from 49.04% to 62.08%, and hence the performance of the system was enhanced and has reached an F-measure of 65.91.

Table 1: Baseline results

Baseline	Precision	Recall	F-measure
Location	75.71%	76.97%	76.34
Misc	22.91%	34.67%	27.59
Organisation	52.80%	33.14%	40.72
Person	33.84%	14.76%	20.56
Overall	**51.39**%	**37.51**%	**43.36**

Table 2: ANERsys 1.0 results

ANERsys 1.0	Precision	Recall	F-measure
Location	82.17%	78.42%	80.25
Misc	61.54%	32.65%	42.67
Organisation	45.16%	31.04%	36.79
Person	54.21%	41.01%	46.69
Overall	**63.21**%	**49.04**%	**55.23**

Table 3: ANERsys 2.0 results

ANERsys 2.0	Precision	Recall	F-measure
Location	91.69%	82.23%	86.71
Misc	72.34%	55.74%	62.96
Organisation	47.95%	45.02%	46.43
Person	56.27%	48.56%	52.13
Overall	**70.24**%	**62.08**%	**65.91**

Impact of Tokenization In our previous works, the error-rate induced by the complex morphology of the Arabic language was not taken into consideration. This error-rate is mainly due to the bad training which is a direct consequence of the sparseness of data caused by the agglutinative morphology. In this paper, we have conducted experiments before and after the tokenizing the data. In Table 4 we present the results obtained with raw text, whereas the results obtained after the tokenization, are presented in Table 5, using CRF (we have used CRF++[12]).

Table 4: CRF results using non-tokenized data

CRF Raw	Precision	Recall	F-measure
Location	95.09%	70.02%	80.65
Misc	78.31%	50.39%	61.32
Organisation	85.27%	46.51%	60.19
Person	80.18%	36.73%	50.38
Overall	**89.20**%	**54.63**%	**67.76**

Table 5: CRF results using tokenized data

CRF Tok.	Precision	Recall	F-measure
Location	95.38%	76.14%	84.68
Misc	79.49%	47.33%	59.33
Organisation	86.28%	48.28%	61.92
Person	84.87%	38.18%	52.67
Overall	**90.82**%	**57.83**%	**70.67**

Features The rest of the tables show the results obtained using each of the features individually and then combining all of them. Table 6, 7, 8 and 9 show the impact of the POS-tag, BPC, GAZ and NAT, respectively.

[11] http://ar.wikipedia.org

[12] http://crfpp.sourceforge.net/

Table 6: Results obtained using the POS-tag feature

POS	Precision	Recall	F-measure
Location	89.88%	86.49%	88.15
Misc	77.91%	51.15%	61.75
Organisation	83.02%	53.33%	64.94
Person	79.29%	65.42%	71.69
Overall	**85.28%**	**71.82%**	**77.97**

Table 7: Results obtained using the BPC feature

BPC	Precision	Recall	F-measure
Location	95.97%	77.28%	85.62
Misc	80.25%	49.62%	61.32
Organisation	85.87%	49.09%	62.47
Person	86.39%	41.52%	56.09
Overall	**91.35%**	**59.62%**	**72.15**

Table 8: Results obtained using the GAZ feature

GAZ	Precision	Recall	F-measure
Location	94.36%	79.21%	86.12
Misc	81.58%	47.33%	59.90
Organisation	85.66%	48.28%	61.76
Person	84.94%	43.66%	57.67
Overall	**90.22%**	**60.85%**	**72.68**

Table 9: Results obtained using the NAT feature

NAT	Precision	Recall	F-measure
Location	95.60%	76.32%	84.88
Misc	79.75%	48.09%	60.00
Organisation	84.86%	48.69%	61.87
Person	85.80%	40.32%	54.86
Overall	**90.83%**	**58.66%**	**71.29**

Table 10: Results obtained combining “all” the features

ALL	Precision	Recall	F-measure
Location	93.03%	86.67%	89.74
Misc	71.00%	54.20%	61.47
Organisation	84.23%	53.94%	65.76
Person	80.41%	67.42%	73.35
Overall	**86.90%**	**72.77%**	**79.21**

7. Results Discussion and Error Analysis

By Features : When each feature was used individually, the POS-tag (Table 6) feature showed the best improvement in F-measure (more than 7 points). The contribution of the POS-tag feature was mainly on the recall (amost 14 points), whereas for the precision it has caused a significant decrease (more than 5 points). The only feature which showed to help increasing the precision is the BPC feature (Table 7). However, the improvement in both precision and recall was very light. Using external resources has only helped to increase 3 points in recall (Table 8), whereas for the NAT feature, it has contributed with an improvement of 0.62 points (Table 9).

By Classes : The CRF model has benefited from all the features for all the classes. However, the results tables show that all the classes have benefited more from the POS-tag feature than the other features on the recall and F-measure levels. On the other hand, the “Location”, “Organization” and “Person” classes show that they gain more in precision with the BPC feature, whereas the “Miscellaneous” class improves more in precision with the GAZ feature. The major difference between the “Miscellaneous” class and the other classes is that the contexts in which its potential subclasses (weapons, currencies, vehicles, etc.) might appear are very different. On the other hand, the NEs which belong to the other classes are more precisely defined and even though they have sub-classes (Person: president, actor, etc. Location: country, city, street, etc. Organization: research center, soccer team, fashion label, etc.) they tend to appear in the same context. For this reason, the “Miscellaneous” class benefits more from using external resources than using other features.

Combination of the Features : When all the features were combined (Table 10), the obtained recall (72.77%) was almost one point above the best recall obtained by a single feature (71.82%, see Table 6), whereas the precision was (86.90%) almost 4 points below the best precision obtained when the BPC feature was used individually (91.35%). However, on the F-measure level, Table 10 shows that the performance is almost 2 points above using only the POS-tag feature. That is, when a CRF model is user with independent features of different types in the NER task, it succeeds to combine these features and obtain results which outperform the ones obtained when these features are used individually.

8. Conclusions and Further Work

In this paper we present our preliminary experiments which aim at improving ANERsys, our NER system for Arabic text, by using the CRF model.

The results showed that with the CRF model we can obtain a performance almost two points higher with respect to the second version of ANERsys which relies on a 2-step approach and partially on a Maximum Entropy model. Due to the complex morphology of the Arabic language, we have performed a tokenization on our data which helped to gain almost three points. Thereafter, we have performed experiments using four different gazetteers individually and combining them. The results showed that we have obtained more improvement in recall than in precision. Moreover, some classes (“Miscellaneous”) showed that they benefit more from using external resources than morphological (POS-tag) feature. When all the features were combined, the CRF models showed that it outperfoms other probabilistic model in the ability to capture arbitrary, overlapping features (Kristjansson et al., 2004). The overall F-measure was enhanced more than one point above the best result obtained using only one feature (POS-tag), almost 9 points above the results obtained when no features were added and almost 14 points above the results obtained with the second version of our Arabic NER system (65.21). All the features that we have used in our experiments are language-independent which will allow many NLP researchers to benefit from our research work for othe languages.

In the next future we plan to increase the size of ANERcorp in order to obtain a higher performance of the system. We also plan to carry out experiments using different feature-sets, and explore the possibility of designing a feature-set for each class. Furthermore, we plan to conduct a comparative study between many probabilistic models (SVM, HMM, Maximum Entropy, CRF, etc.) and also experiments using a combination of different models.

Acknowledgments

The research work of the first author was partially supported by MAEC - AECI. We would like to thank the PCI-AECI A/010317/07 and MCyT TIN2006-15265-C06-04 research projects for partially funding this work.

9. References

Abuleil S. 2002. *Extracting Names from Arabic text for Question-Answering Systems. Computers and the Humanities.*

Babych B. and Hartley A. 2003. *Improving Machine Translation Quality with Automatic Named Entity Recognition.* In *Proc. of EACL-EAMT*. Budapest.

Benajiba Y., Rosso P., Benedí J.M. 2007. *ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy..* In *In: Proc. 8th Int. Conf. on Comput. Linguistics and Intelligent Text Processing, CICLing-2007, Springer-Verlag, LNCS(4394), pp. 143-153..*

Benajiba Y., Rosso P. 2007. *ANERsys 2.0 : Conquering the NER task for the Arabic language by combining the Maximum Entropy with POS-tag information..* In *In: Proc. Workshop on Natural Language-Independent Engineering, 3rd Indian Int. Conf. on Artificial Intelligence, IICAI-2007, Pune, India, December 17-19.*

Benajiba Y., Rosso P. and Lyhyaoui A. 2007, *Implementation of the ArabiQA Question Answering System's Components, In Proc. of ICTIS-2007*,

Bender O., Och F., and Ney H.. 2003. *Maximum Entropy Models For Named Entity Recognition.* In *Proceedings of CoNLL-2003*. Edmonton, Canada.

Chieu H. and Ng H. 2003. *Named Entity Recognition with a Maximum Entropy Approach.* In *Proceedings of CoNLL-2003*. Edmonton, Canada.

Cucerzan S. and Yarowsky D. 1999. *Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence.* In *Proceedings, 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pp. 90–99.

Curran J. R. and Clark S. 2003. *Language Independent NER using a Maximum Entropy Tagger.* In *Proceedings of CoNLL-2003*. Edmonton, Canada.

Ferrndez S., Ferrndez O., Ferrndez A. and Muoz R., 2007. *The Importance of Named Entities in Cross-Lingual Question Answering, In Proc. of Recent Advances in Natural Language Processing, RANLP-2007.* Borovets, Bulgaria.

Greenwood M. and Gaizauskas R. 2007. *Using a Named Entity Tagger to Generalise Surface Matching Text Patterns for Question Answering, In Proceedings of the Workshop on Natural Language Processing for Question Answering (EACL03)*,

Kristjansson T., Culotta A., Viola P., and McCallum A. 2004. *Interactive Information Extraction with Constrained Conditional Random Fields.* In *Proceedings of AAAI-2004.*

Kudo T., Yamamoto K., and Matsumoto Y. 2004. *Applying Conditional Random Fields to Japanese Morphological Analysis.* In *Proceedings of EMNLP*, 2004.

Lafferty J., McCallum A., and Pereira F. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.* In *Proceedings of the Eighteenth International Conference on Machine Learning*. 2001.

Li W. and McCallum A. 2003. *Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction.* In *Special issue of ACM Transactions on Asian Language Information Processing: Rapid Development of Language Capabilities: The Surprise Languages.*

Maloney J. and Niv M. 1998. *TAGARAB, A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis.* In *Proceedings of the Workshop on Computational Approaches to Semitic Languages.*

Malouf R. 2003. *Markov Models for Language-Independent Named Entity Recognition.* In *Proceedings of CoNLL-2003.* Edmonton, Canada.

McCallum A. and Li W. 2003. *Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.* In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL).*

Mollá D., van Zaanen M. and Smith D. 2006. *Named Entity Recognition for Question Answering, Proc. of the Australasian Language Technology Workshop Sancta Sophia College*

Pinto D., McCallum A., Wei X., and Croft W. B. 2003. *Table Extraction Using Conditional Random Fields.* In *Proceedings of the 26th ACM SIGIR.*, 2003.

Settles B. 2004. *Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets.* In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA).*

Sha F. and Pereira F. 2003. *Shallow parsing with conditional random fields.* In *Proceedings of HLT-NAACL.*

Strótgen R., Mandl T. and Schneider R. 2005. *A Fast Forward Approach to Cross-Lingual Question Answering for English and German.* In *Proceedings of the Workshop of Cross-Language Evaluation Forum (CLEF).* 2005.

Thompson P. and Dozier C., 1997. *Name Searching and Information Retrieval, In Proc. of Second Conference on Empirical Methods in Natural Language Processing*,

Toda H. and Kataoka R., 2005. *A Search Result Clustering Method using Informatively Named Entities, In Proc. of the 7th annual ACM international workshop on Web information and data management.*,

Tran Q. T., Pham T. X. T., Ngo Q. H., Dinh D., and Collier N. 2007. *Named Entity Recognition in Vietnamese documents. Progress in Informatics Journal.* 2007.

Wu C-W., Jan S-Y., Tsai R. T-H., and Hsu W-L. 2006. *On Using Ensemble Methods for Chinese Named Entity Recognition. Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing.* 2006.

Can the building of corpus-based Arabic concordances with AraConc and DIINAR.1 tackle the issue of Arabic polyglossia?

Joseph Dichy, Ramzi Abbès
Université Lumière-Lyon 2 / ICAR (CNRS-Lyon 2)
Faculty of Languages, 74 rue Pasteur – 69365 Lyon Cedex 07
E-mail: joseph.dichy@univ-lyon2.fr, ramzi.abbes@univ-lyon2.fr

Abstract

Considering the high level of linguistic variation commonly observed in the Arabic language as a whole, including Arabic vernaculars (or ‘dialects’), the real challenge for to-day’s NLP is both to keep hold of the current sate of knowledge in the description of linguistic variation, and at the same time, to seek efficient formalized approaches that allow software applications to operate at a sufficient level of granularity. In this paper, we try and show how a reasonable level of granularity and of subsequent feasibility can be reached through a balance between the complexity of Arabic corpus data and present-day tools that have been originally devised for Modern Standard Arabic (MSA). In view of the complex system of the communicative competence in Arabic (and the concept of Arabic *polyglossia*), we suggest a primary mapping of what could become, in the future, a real-world hyper-base of corpora in the Arabic language, then endeavour to contribute to the above challenge, through the use of tools initially devised for MSA(the DIINAR.1 knowledge database [*DIctionnaire INformatisé de l’ARabe, version 1*] and the AraConc concordance software. Two types of actual results are made available: statistical results and concordances. Examples of how these tools can – or not – meet linguistic variation in Arabic are given

1. Introduction

Upon considering present-day Arabic NLP software and resources from a ‘SWOT analysis’ standpoint, one can hardly avoid detecting a ‘weak point’ related to an oversimplified view of the Arabic language, when confronted to the very high level of linguistic variation commonly observed, and even more commonly denied, except when it comes to Arabic vernaculars (or ‘Dialects’). There is often a gap between present-day research and development in Arabic NLP, where oversimplifying views are often held, and the results of over a century of modern linguistic research and description in the field of Arabic language variation and dialectology. The real challenge is both to keep hold of the current sate of knowledge in the description of linguistic variation – complex as it may seem, it is in itself a simplified representation when compared to real-world language use –, and at the same time, to seek efficient formalized approaches that allow software applications to operate at a sufficient level of granularity.

In this paper, we try and show how a reasonable level of granularity and of subsequent feasibility can be reached through a balance between the complexity of Arabic corpus data and present-day tools that have been originally devised for Modern Standard Arabic (MSA).

Section 1 introduces the complex system of the communicative competence of Arabic speakers, and recalls some of the research that have been done, on the whole, since the second world war, on Arabic variation.

Section 2 tries and answer the open questions of section 1, through suggesting a mapping of what could become, in the next future, a real-world hyper-base of corpora in the Arabic language. The question is also, in flat words, that of what the phrase ‘Arabic language’ refers to in scientific terms, and of how the challenge of applying NLP techniques to real-world Arabic can be met.

Section 3 is an endeavour at contributing to the above challenge, through the use of tools initially devised for MSA. These are the DIINAR.1 knowledge database (*DIctionnaire INformatisé de l’ARabe, version 1*) and the last born of the analyzers based on it, the AraConc concordance software, which has been devised to extract concordances and frequency lists in Modern Standard Arabic by R. Abbès (PhD, supervised by J. Dichy and Mohamed Hassoun). The functions included in the software, which are based on the specific structures of Arabic texts (many of which are shared by Semitic languages of the same group) allow building concordances of a very high level of accuracy (words are recognised whatever clitics or prefixes are added, and all the words pertaining to a given root can be given). Two types of actual results are made available: statistical results and concordances. Examples of how these tools can meet linguistic variation in Arabic, and begin to evolve towards a new generation of tools are given.

2. The issue of Arabic variation and the concept of *polyglossic languages*

The computational linguistics of Arabic, which try and answer the requirements of everyday users of new technological software and hardware communication devises, and industrial needs related to real speech analysis and generation, are confronted to one of the most challenging issues in Arabic linguistics in general, that of linguistic variation. It is well known that there are a great number of Arabic varieties, traditionally described as Arabic dialects or vernaculars, in addition to classical Arabic, Modern Standard Arabic and the use of ‘medium’ or ‘mixed varieties’, which have been observed since the first centuries of Islam, i.e. as early as the 7th-10th centuries c.e., and onwards – see (Blau, 1961, 1977, 2002), and also, among many others: (Hopkins 1984), (Doss 1991), (Larcher 2001, 2003), (Lentin, 2008).

Ferguson (1959), after Marçais (1930), described Arabic as ‘diglossic’. The notion of diglossia has, in the course of time, been shown to be too narrow for Arabic – see the data of (Badawi, 1973), (Doss 1987), (Bani Yassin and Owens, 1987a, b), (Owens and Bani Yassin, 1991), (Burési, 1991), (Medfai, 1998), and also Ferguson’s ‘Diglossia revisited’ (1991). The need for a concept that could be at the same time narrower than to be used for describing different languages in contact – as had been proposed by Fishman (1967) –, and meet the complex structure of the communication competence in the Arabic language has emerged (Youssi, 1983), (Dichy, 1987).

The concept is that of *polyglossia* (Dichy, 1994, 2007). In this view, the linguistic knowledge system of the average schooled Arabic speaker results from the fact that a wide set of similarities on the one hand, and a narrower set of dissimilarities on the other, can be observed between the language acquired at home (the 'home language') and that acquired at school (the 'school language'). The former is related to the 'Arabic Dialects' (or 'vernaculars') which vary from one region to another, and the latter, both to classical (or medieval) Arabic and to what is currently called Modern Standard Arabic.
The relation between the first and the second is submitted to ideological pressure from two main sides, which can be described, in very rough words, as follows:

(a) the Arab nationalist view rejects any reference to Arabic vernaculars in the teaching of the language, and constantly mixes up between classical and modern standard Arabic;
(b) the other position holds that there as many languages as there are Arabic Dialects. This implies, explicitly (Martinet, 1966) or not, that Arabic would be a group of languages.

In the author's opinion (Dichy, 2003), both sides potentially lead the scientific community astray. This view is strengthened by many of the references quoted above. The similarities/dissimilarities between 'home' and 'school' languages and many other facts that can be observed in Arabic communication behaviour everywhere in the Arab world show that what such common phrases as 'speaking Arabic', 'an Arabic film' (or 'song'), 'communication in Arabic', etc., refer to a complex system, which can be represented in the paragraph below.

2.1. The components of Arabic polyglossia

In the table below, which is taken up, with modifications, from (Dichy, 1994), variation observed in Arabic is considered from a cognitive standpoint, i.e., from that of the knowledge system of a schooled speaker of Arabic. This is another viewpoint than the now traditional sociolinguistic approach, albeit the two ways of considering these phenomena are not contradictory.

Glossa	Short definition
Standard Arabic	
• **Classical Standard Arabic** (CSA)	CSA is the glossa found in most Medieval Arabic texts, the most ancient of which are pre-Islamic poetry and the Koran. It is nowadays strictly used in recognition skills (or in reproduction skills, with or without previous learning by heart). CSA is the *historical reference glossa* shared by the Arab World as a whole. Its acquisition is in fact restricted to understanding skills, and only occurs in schools. It is therefore part of the 'school language'.
• **Modern Standard Arabic** (MSA)	MSA is the contemporary state of CSA. It is the glossa of medias, administration, contemporary literature, sciences and technical development (etc.). MSA the *geographical reference* shared by the Arab World as a whole. Its acquisition occurs both in expression and understanding skills. It is the central glossa of the 'school language'
Medium Arabic	
• **TYPE 1** (MA-1)	The basic descriptive hypothesis presented in Dichy (1987, 1994), among others, is that there are two main types of Medium Arabic, which 'mixes' structures from MSA and from the MRV (see below) of a given large region of the Arabic speaking world. MA-1 is – very roughly – produced through inserting words and phrase belonging to the 'regional' vernacular of the speaker in sentences the syntax of which globally pertains to MSA.
• **TYPE 2** (MA-2)	MA-2 in the above basic hypothesis can be described as the result of the insertion of words and phrases from MSA in sentences, the syntax of which belongs on the whole to the speaker's MRV. The main consequence of this analysis is that Medium Arabic is neither an 'inter lingua' nor a 'lingua franca'. Nor can it be brought to the status of a shared 'Educated Spoken Arabic' (Mitchell, 1986). *The two types of Medium Arabic vary from one Arabic speaking region to another.*
Arabic Vernacular (or 'dialect')	
• **'Regional'** **(**Egy-ARV, Syr-ARV, Mor-ARV...)	The 'regional' vernacular is the *reference spoken glossa of a given 'region'* (*'iqliim* إقليم) of the Arab World: Jor-ARV, Tun-ARV, etc., refer respectively to Jordanian ARV, Tunisian ARV, etc. The reference glossa is often related in a given country, to the prestige of the vernacular used in the Capital city. The determination of ARV-s nevertheless remains conventional.
• **Local** (ALV)	The ALV is, for a given speaker, the vernacular in use in the part of a large city, or in the village, valley, etc. he dwells in
'Neighbouring' Arabic Local Vernaculars	
NALV^1, NALV^2, NALV^3...	NALV-s are Local Vernaculars other than that of the speaker, but belonging either to the same 'region', or, in a given 'region', to the same group of vernaculars. The level of understanding is high.
Other Arabic Regional Vernaculars	
'other ARV'	'Other ARV-s' come from other 'regions' of the Arab World that that of the speaker. The latter is often – and even the more to-day, due to the development of satellite TVs – brought to develop a certain level of understanding skills in other Regional Vernaculars. The level of understanding varies.

Table 1: The components of a schooled speaker of Arabic, i.e. of Arabic polyglossia

The table above has been considered relevant enough to be included in French school programmes for the teaching of Arabic in Grammar schools (BOEN, 1995; 1997; CNDP, 1996, 1997).

3. Can the issue of Arabic variation be met with NLP software and languages resources?

The above table can be used as an overall scheme for the mapping of a new generation corpora and also for the idea of a hyper-lexical resource. These are two conditions for a positive answer to the title question of this section.

How can the whole set of data covered by the phrase 'the Arabic language' can be mapped? The scheme is three dimensional (Dichy, 2003):

- The horizontal axis correspond to the geographical extension of the Arabic language, and to the associated glossas and variations.

- The vertical axis is that of the diachronic evolution of the language, which includes the evolution of Arabic glossas from the 1st century of Islam (7th cent. c.e.) until now, as they can be reached for every country of the Arabic speaking world throughout History. The overall periods of the Arabic Lexicon have been sketched as encompassing (Dichy, 1998):
  (a) the 'language of the Ancient Arabs' (*lisaan al-'Arab*), which includes the original ancient lexicon ;
  (b) the vocabulary of medieval Arabic civilization, which contains items referring to the life and institutions of cities and to Islamic civilization, from the first centuries of the Hijra to the 18th century c.e.;
  (c) the vocabulary of the modern age, the starting point of which is traditionally situated at the dawn of the 19th century, i.e. at the beginning of the *NahDa*, the Arab 'Renaissance'.

- The third axis considers the communicative competence of a given learned (or, in Modern times, schooled) speaker of Arabic presented in § 2.1. The components of table 1 vary across the Arab world (except for CSA and MSA, albeit a certain level of lexical variation is also encountered in MSA).

The idea is that corpora can be indexed in relation to the above mapping.

4. Can LR-s and software devised for MSA be used for other Arabic glossas?

Clearly, tools and resources devised for MSA are only partly applicable to other components of Arabic pluriglossia. The further towards vernaculars (or 'dialects') one needs to go, the more difficult the task. The recognition process, on the other hand, is easier in the written realizations that it with sound. Writing includes in every language where it is found, a certain level of abstraction and convention. In Arabic, three or four-consonants roots are shared by different glossas, regardless of geographic variation, to a good extent.

There also are written texts – albeit in variable quantity – in every Arabic glossa (Blau, 1961, 2002). This means that, provided one sticks, for a start, with written texts, the results of applying tools devised for MSA to other glossas can be considered as a first stage, and evaluated.

4.1. Results obtainable through concordances built with AraConc and DIINAR.1, and the evolution of tools originally devised for MSA

AraConc (Abbes, 2004; Abbes and Dichy, 2008) is a concordance software based on the DIINAR.1 (DIctionnaire INformatisé de l'ARabe, version 1) lexical resource, and the related morphological analyzers (Dichy and Hassoun, 2005 – also: http://diinar.univ-lyon2.fr; availability of DIINAR.1: www.elda.org).

The concordance software analyses every word-form in the text, and associates it with it its context through:
– showing around 10 words right and left of the word-form,
– indicating the part of the corpus where the word-form has occurred.

This classical concordance presentation is preceded by a thorough morphological analysis of the word-form (Dichy, 1997; Abbès, 2004), which is both lemmatized and related to its root. This allows two types of interrogation:
– lemma query,
– root query.

Arabic word-forms in a concordance cannot be dissociated from their analysis. They need to be related to their root on the one hand, and to keep their clitics and affixes on the other. Multiple interpretations and analyses are to be distinguished by the user. In order to ensure interaction with users, the position of every word-form in the corpus is, as indicated, memorised. (Abbès and Dichy, 2008). Thus, the output of the software is a triple: the word-form, its analysis (lemmatization, root relation, inclusion of clitics and affixes) and its position in the corpus.
In addition, AraConc allows statistics based on word-forms (lemmatization and the resolution of word-form ambiguities can only be achieved, in the present state of the art, through human analysis).

The issue of what can be done with tools originally built for MSA to analyse texts produced in other glossas, especially regional or local vernaculars, raises two questions: (a) What parts of the above scheme (§ 3 and table 1) can be tackled with such tools and resources? (b) Can new tools and resources be developed with the help of what has been built for MSA, and on what basis?

4.2. What existing tools and resources allow

The DIINAR.1 lexical resource is based on both CSA and MSA: the set of dictionaries that were compiled for lexical information included, in addition to MSA, a comprehensive coverage of Medieval Arabic. This results in the fact that AraConc can be used to build concordances based on medieval corpora, in addition to contemporary texts. Such concordances allow studying the diachronic evolution of the vocabulary and of the associated grammar-lexis structures and collocations.

Let us consider a few examples:

(1) The quantifier *bid'*بضع, meaning 'a small number of' (classically: between 3 and 10), is followed in a good percentage of its occurrences – a rough 55% – in contemporary newspapers and novels by words referring to time units, such as *'ayyaam* أيام, 'days', *'ašhur/šuhuur* شهور/أشهر , 'months', *siniin* سنين , 'years', etc., followed, especially in newspapers, by words expressing numbers, *'aalaaf* آلاف, 'thousands', *'ašaraat* عشرات, 'tens', etc. (the percentage is around 22%). In the *Kitaab al-Hayawaan* ('The Book of Living creatures') of Al-JaaHiZ (8th cent. c.e. century), such percentages do not appear at all. The occurrences do not appear to be followed by any given category of words.

(2) What would be expressed in English by 'lest', 'for fear that' (followed by a verbal proposition) often appears, in MSA, as *xawfan min 'an* خوفا من أن . In CSA (as exemplified, our corpus, by the works of Al-JaaHiZ), one either finds *maxaafata 'an* مخافة أن or – it seems, less frequently – *xawfan'an* خوفا أن. The former is also found in MSA. In the latter, the preposition *min,* 'of' or 'from', which appears in MSA, is omitted.

(3) A newly appeared word, *šaxSana* شخصن, 'to personalize' (= to turn into a matter directly related to a given personality, as in the French word 'personnaliser', which is not the exact English counterpart) could be found in the second half of the 1990's in a Lebanese newspaper such as *Al-Hayaat,* but not in the Egyptian daily *Al-Ahraam*. We did not find it in novels (our corpus does not include, presently, very recent works): this may be due to the political sense taken by *šaxSana*, which explains its appearing in newspaper writing. Of course, the verb did not occur in CSA.

(4) The verb *taraka* ترك, 'to leave', 'quit', 'abandon' is always followed by complements in CSA, and almost always in MSA. The use of *taraka* without any complement (not event an implicit complement, the cue of which is given in the context), meaning, 'he left', 'he quitted', may also appear in MSA, in Levantine novels or newspapers. The influence of the regional vernaculars is probably to be observed here.

The analysis of the examples above has been shortened and simplified. These, and quite a few other ones, show that concordances built with MSA-based tools allow:

- contrasting CSA and MSA, which is a crucial issue with respect, especially, to education (examples 1, 2 and 4);
- observing variation between newspaper writing and novels (examples 4);
- observing variation in MSA, between different Arab countries (examples 3 and 4);
- observing the appearing of new words and usages (example 4).

It is to be noted that the tools we have used have directed us towards lexical variation, including grammar-lexis variation (example 4 illustrates a difference in the argument structure of a verb). Some of the observations are of purely collocative nature (example 1).

5. Conclusion: future resources, concordance software and lexica

This brings us back to question (b) above: can new tools and resources be developed with the help of what has been built for MSA, and on what basis?

The above tools prove efficient for searching and analysing data in MSA, and, thanks to the very wide lexical coverage of DIINAR.1, in CSA.

They allow, in addition, the building of a new generation of lexical resources, that will include semantic relations, in addition to the morphosyntactic information already included in the specifiers of DIINAR.1 (Dichy, 1997).

What will now need to be added is a software aimed at the recognition of tool-words (*'adawaat*, أدوات) in MSA and CSA. Such a software should include contextual analysis grammars.

There is also a need for a new set of morphological analyzers that take into account the variation observed in the numerous vernaculars of Arabic, and draw grammar-lexis information from a new set of lexical databases. Language resources (corpora and lexica) in Arabic need to take up the challenge of new extensions.

6. Acknowledgements

The present contribution is included in the participation of the Lyon 2-Lumière University and the ICAR CNRS/Lyon 2 Lab to the MEDAR project, supported by the European Community, and coordinated by the Center for Sprogteknologi (Prof Bente Maegaard), University of Copenhagen, www.cst.dk.

References

Abbès R. (2004). *La conception et la réalisation d'un concordancier pour l'arabe*. PhD, Lyon : INSA.

Abbès R. & J. Dichy (2008). "Extraction automatique de fréquences lexicales en arabe et analyse d'un corpus journalistique avec le logiciel AraConc et la base de connaissances DIINAR.1", in : Serge Heiden & Bénédicte Pincemain, *Proceedings of JADT 2008, 9th International Conference on Textual Data statistical Analysis,* Lyon 12-14.03.2008, Presses Universitaires de Lyon, 2 vol. : 31-44.

Abbès R., J. Dichy and M. Hassoun (2004). "The Architecture of a Standard Arabic Lexical database: some figures, ratios and categories from the DIINAR.1 source program". *COLING 2004*, Geneva.

Badawi, S.M. (1973), *Mustawajaat al-'arabijja l-mu'aaṣira fii miṣr* ("The levels of contemporary Arabic in Egypt"), Cairo : Daar al-ma'aarif.

Bani Yasin R. and Owens, J. (1987a), *Variation in Rural Northern Jordanian Arabic,* Irbid (Jordanie) : Yarmouk University Publications.

— (1987b), *The Lexical Basis of Variation in Arabic,* Irbid (Jordanie) : Yarmouk University Publications.

Blau, J. (1961). "The Importance of Middle Arabic Dialects for the History of Arabic". Heyd, U. ed.

Studies in Islamic History and Civilization. Scripta Hierosolymitana IX, Jerusalem: Magnes-Hebrew University. 206-228.

— (1977). “The Beginnings of the Arabic Diglossia: A Study of the Origins of Neoarabic”. *Monographic Journals of the Near East*. Reprint from Afroasiatic Linguistics, 4, 175-202.

— (2002). *A Handbook of Early Middle Arabic.* Jerusalem, The Hebrew University Press.

BOEN, Bulletin officiel du ministère de l'Éducation nationale (1995), *Programmes de langues vivantes pour le nouveau collège,* décembre 1995 : "Programmes de langue arabe : Orientations pour l'ensemble du collège et pour la sixième LV I " ;

— (1997), "Programme de langue arabe - classes de cinquième et de quatrième LV I", hors-série n°1 du 13 février 1997, vol. 1, pp. 97-99.

Burési M. (1991), *Recherches sur l’Arabe parlé en Tunisie à partir de Causeries radiodiffusées (1970-1971) du chroniqueur Adbellaziz LAROUI,* Thèse de Doctorat, Université Paris III, 3 vol.

CNDP (1996), Documents d'accompagnement des programmes de langue arabe: *Accompagnement des programmes de sixième - Lecture transversale et thématique des programmes,* Livret 2 Paris, 1996 (chapitre concernant l'arabe : pp. 38-48).

— (1997) Accompagnement des programmes de *cinquième et de quatrième* - Livret 7 : arabe, espagnol, italien, portugais, russe, CNDP, Paris, 1997, pp. 7-11.

Dichy J. (1987). « Qu’est-ce qu’un programme d’apprentissage de la compétence communicative d'un locuteur arabe scolarisé ? », *Actes du Colloque sur les «Langues et cultures populaires dans le domaine arabe»* (Paris, 16-18 octobre 1986), Paris, Association française des Arabisants (A.F.D.A.) et Institut du Monde arabe, 49-61.

— (1994). « La pluriglossie de l'arabe », P. Larcher (ed.) special issue of the *Bulletin d'Études orientales,* Institut français d'Études arabes de Damas (IFEAD), XLVI, pp. 19-42.

— (1997). “Pour une lexicomatique de l’arabe : l’unité lexicale simple et l’inventaire fini des spécificateurs du domaine du mot”. *Meta* 42, Québec, pp. 291-306. www.erudit.org/revue/meta/1997/v42/n2/002564ar.pdf

—, 1998, « Mémoire des racines et mémoire des mots : le lexique stratifié de l’arabe », *Revue tunisienne des Sciences Sociales,* n° spécial : V^e Journées scientifiques du réseau “Lexicologie, Terminologie et Traduction” de l’AUPELF-UREF (Tunis, 25-27 sept. 1997), sur la *Mémoire des mots,* pp. 93-107.

— (2003). « La variation linguistique comme fait culturel : l'exemple de l'arabe et de son enseignement en France », in *Les contenus culturels dans l'enseignement des langues vivante,* coll. « Les Actes de la DESCO » (Paris, 4-5 décembre 2003), Ministère de l’éducation nationale, Académie de Versailles : CRDP, p. 79-101. http://eduscol.education.fr/D0126/contenus_culturels_dichy.htm

— (2007). “La pluriglossie de l’arabe en (inter)action : un exemple conversationnel syrien”, in Baudoin DUPRET, Zouhair GHAZZAL, Youssef COURBAGE et Moahmmed AL-DBIYAT (éd.), *La Syrie au présent*, Paris, éditions Actes-Sud/Sinbad, p. 495-505.

Dichy, J. and M. Hassoun (2005). “The DIINAR.1-« مـعـالـي » Arabic Lexical Resource, an outline of contents and methodology”, in *The ELRA Newsletter*, Vol. 10, n°2, April-June 2005, pp. 5-10.

DIINAR.1 web site: http://diinar.univ-lyon2.fr

Doss, M. (1987), « Les variétés linguistiques en usage à la télévision égyptienne », in *Bulletin du CEDEJ*, n°21, Le Caire, 63-74.

— (1991). *L’arabe en Égypte. Étude évolutive d’une langue de relation.* Thèse de doctorat, Université de Paris III.

Ferguson C. (1959a), “Diglossia”, in *Word*, vol. 15, 325-340 (reprod. in Hymes, D. (éd.), *Language in Culture and Society*, New-York : Harper & Row - London: Evanston, 1964, 429-439).

— (1959b), “Myths about Arabic”, in R.S. HARRELL (ed.), *Georgetown University Monograph Series on Languages and Linguistics* 12, p. 75-82 (repr. in J.A. FISHMAN, ed., *Readings in the Sociology of Language,* Mouton, La Hague, 1968, p. 375-81).

— (1991), “Epilogue : diglossia revisited”, *Southwest Journal of Linguistics*, 10/1, 214-34 (reprod. in A. Elgibali, ed., *Understanding Arabic, Essays in Contemporary Arabic Linguistics in Honor of El-Said Badawi,* The American University of Cairo Press, 1996, 49-67).

Hopkins, S. (1984). *Studies in the Grammar of early Arabic*. Oriental Series, v. 37, Oxford: Oxford University Press.

Larcher, Pierre (2001). “Moyen arabe et arabe moyen” dans *Linguistique arabe : sociolinguistique et histoire de la langue (sous la dir. de P. Larcher)*, *Arabica* 48/4 : 578-609. Leiden : Brill.

— (2003). “*’ayy(u) šay’in, ’ayšin, ’ēš* : moyen arabe ou arabe moyen ?”, *Quaderni di Studi Arabi* 20-21, 2002-2003, p. 63-78.

Lentin, J. (1997). *Recherches sur l’histoire de la langue arabe au Proche-Orient à l’époque moderne,* thèse de doctorat d’État, Université Paris III, 2 vol.

— (1998). “Middle Arabic”, *Encylopaedia of Arabic Language and Linguistics,* K. Versteegh, dir., Leiden : Brill, vol. 3, p. 215-224.

Martinet, A. (1966), “Bilinguisme et plurilinguisme” – “Hiérarchie des usages linguistiques” - “Les langues dans le monde de demain”, in *Revue tunisienne de Sciences Sociales,* n°8, Tunis, pp. 57-77, 103-114, 165-173.

Medfai, A. (1998), *Réalisations tunisiennes de l’arabe moyen, à partir d’un corpus télévisé,* thèse de Doctorat en Sciences du Langage, Université Lumière-Lyon 2, 2 vol.

Mitchell, T.F. (1986). “What is Educated Spoken Arabic?” *International Journal of the Sociology of Language* 61, pp. 7-32.

Owens, J. and Bani Yasin R. (1991). “Spoken Arabic and Language Mixture”, in P. Larcher (ed.) special issue of the *Bulletin d'Études orientales,* Institut français d'Études arabes de Damas (IFEAD), XLVI: 17-31.

Youssi A. (1983), “La triglossie dans la typologie linguistique”, in *La Linguistique*, n°19, 1983-2, Paris : P.U.F.

Amazigh Data Base

El Mehdi IAZZI, Mohamed OUTAHAJALA
Institut Royal de la Culture Amazigh
Avenue Allal El Fassi, Madinat Al Irfane - Rabat - Instituts Adresse postale : BP 2055 Hay Riad Rabat
E-mail: iazzi@ircam.ma, outahajala@ircam.ma

Abstract

This paper focuses on the linguistic and computational aspects of a project which the Royal Institute for Amazigh Culture is carrying out .The project deals with the elaboration of an application that will help in collecting and accessing Amazigh words. The objective is to present some aspects of the Moroccan Amazigh data base project, its structure and the processing of variations within the framework of the linguistic norm elaborated in Morocco. The application modelling and computing tools used are also presented here. In order to optimise its use, the data base of the Amazigh language is conceived from the beginning in such a way as to provide all the necessary information for each entry. Furthermore, this information allows interrogating the data base from different angles: the classification of the glossary by domains (for example, agricultural glossary, handcraft glossary….), the derivational families such as words derived from the same root, Arabic French and English equivalent words, etc.

1. Introduction

Data processing has become a tool of teaching whose importance one cannot deny. This is reflected both in the field of teaching and dictionaries. In this global context Royal Institute for Amazigh Culture, known as IRCAM has elaborated an application that will help in collecting and accessing Amazigh words.

The objective of this project is to accompany the integration of the Amazigh in the educational system and the Moroccan Media putting at the disposal of teachers and users of the Amazigh language a data base that should satisfy needs.

For reasons of clarity, we structured this article as follows: the first part will present the context and the expected results. After, we will present the linguistic needs for dictionary that goes in conformity with Moroccan Amazigh standard as taught in the Moroccan school.The third and fourth parts will be about the conception of the computer solution and its realization.

2. The context and the expected results

In this section, we will present the Amazigh standardization process as well as the context and the expected results of the realization of this project.

2.1 Amazigh standardization

The standardization of the Amazigh language cannot be achieved without adopting a realistic strategy that takes into consideration the variation and the linguistic diversity (Iazzi et al., 2007; Ameur et al. 2006). On the spelling side, it is based on a system neutralizing the phonetic divergences between the dialects over the geographical limits of the Moroccan state. The most important differences are due to the spirantisation of the plosives and to the assimilations resulting from the juxtaposition of some phonemes. Writing will permit the neutralization of these linguistic problems. An important work has been achieved with the codification of the Tifinaghe in Unicode (Outahajala, 2005; Zenkouar, 2004) and the teaching of the Amazigh in the primary school in Morocco.

2.2 Context and expected results

The Amazigh lexicography published up to now consists of a set of works dedicated to the vocabularies (glossaries or even some dictionaries) or a set of works of grammatical descriptions or the collection of texts and corpora. These publications have the following features:

- They are about some dialects that are circumscribed geographically.
- Some variants are represented less than others, or even not studied.
- A general work regrouping the data of all dialects doesn't exist. The researchers who work in this domain must consult the totality of these publications.
- These publications are scattered and inaccessible in most cases. Some of them go back to the XIXe century and the beginning of the XXe century. The few existing copies are only available in specialized libraries in France (generally).
- General documents regrouping the data of all dialects of Tamazight do not exist (phonetics, semantics, morphology, phraseology…etc.). The researchers who work in this domain must consult several publications.

For these reasons, coupled with technological advances, IRCAM integrated in his action plan the conception and the realization of an electronic dictionary at the disposal of Tamazigh teachers and writers, as well as the researchers and students.

To feed this data base we need, on the one hand, data sources such as existing documentary sources as the glossaries, the lexicons, the texts, the corpora and the grammatical descriptions and on the other hand, of investigations and verifications on field. The written sources will make the object of a systematic deloucing conjugated with verifications and complements.Therefore, this work will constitute the main frame of the data base. This work will continue through a collection program of the lexical set of the different uses of the Amazigh and unpublished oral texts.

3. Amazigh Linguistic needs for the dictionary

In order to take into account structures of the lexicon (formal and semantic structures, convergences and divergences intra and and inter dialectal) and to satisfy the needs of users, we have kept the following aspects:

- The standardized lexical entry, that conforms to the adopted alphabet with a link toward the orthographic rules (aorist/imperative second person of the singular for the verb, singular of basis for the noun, non suppletion for the preposition, etc.)
- The regional phonetic realization(s) [spirantisation, affrication, substitution, deletion, compensatory vocalic elongation, etc.].
- The root of the entry to facilitate the regrouping of word families.
- The grammatical category and the morphological or suppletive variation.
- Common noun / proper noun / noun of relationship… etc. for the noun.
- Gender / number / state of the noun and the adjective;
- Transitivity - valence and conjugation for the verb;
- Suppletive variation for the preposition;
- Values of the determinant/ the preposition/ the adverb/ the conjunction, etc.
- If a word is derived, the nature of the derivation (causative, reciprocal, reflexive, passive etc. for the verb; and action noun, agentive noun, location, instrumental, noun of state) as well as the base of the derivation.
- The geographical linguistic indications: it is about specifying the dialectal zones and the dialects where the word is attested (e.g. Tarifit - Aït Iznassen, Ikebdanen, Tamsaman, Igueznayen, etc. -; Tamazight - Aït Ndhir, Zemmour, Aït Mguild, Izayanen, etc.).
- Source(s) of the word: document published (precisions as to the reference: author, date of edition, page) or collection (to give informant / lexicographer).
- If the word is a borrowing, mention is made on source language (Arabic, French, Phoenician, Hebrew, Latin, Spanish, etc.) and the origin form.
- In case of neologism, mention is made to the source(s) and the complete reference;
- The meaning of the lexical entry in Amazigh and variants of the Amazigh where each meaning is attested (with indications as to source(s): document or informant / lexicographer).
- The synonym (s) in Amazigh to the intra and inter dialectal level (case of one or many meanings) with a link to each synonym.
- The homonym (s) to the intra and inter dialectal level (case of one or many meanings / case of homonymy) with a link to each homonym.
- The antonym (s) to the intra and inter dialectal level with a link to each antonym.
- The equivalents in Arabic, in French, in English…etc.
- The domains of use (agriculture, anatomy, architecture, social structures, etc.).
- The free variation for an entry (with precisions on the regional variants where each free variation is attested, as well as more of a link toward the entry of every free variation.
- Phraseology contexts and frozen structures.
- Illustration(s): photos, sketches, etc. (mainly for the arts, the professions, the techniques, the elements of the natural environment (trees, animals, etc.).

4. Conception of the computer solution

In order to achieve this application we have used UML modeling language. The actors of the application are: the lexicographer, the moderator, the administrator and the anonymous.

4.1 Use cases

The diagram below (Figure 1) shows us actors of the application and its foreseeable use cases implemented in the present version. The anonymous user is the only actor who doesn't need to log in to access the application. He can make simple or advanced researches (by interrogating the data base from different angles: by domains: agriculture, handicraft, media…etc. by root, by dialect…etc., by equivalents…etc.). The lexicographer can manage the lexical entries that he has created (to add, to delete or to edit the lexical entries that he has created and are not validated yet by the moderator), the lexicographer has to fill several forms representing the morphology or the suppletive variations, the meaning(s) of the word, the free variations…etc. He can also comment the words created by the other lexicographers, in the goal of improving the definitions given to the lexical entries. The moderator validates words created by the lexicographers, he can also modify increase all the already entered words, while updating the morphology, the meaning (s), the derived words…etc. He can also parameterize the application by adding dialects, sources, domains…etc. The administrator manages the information about users of this web application.

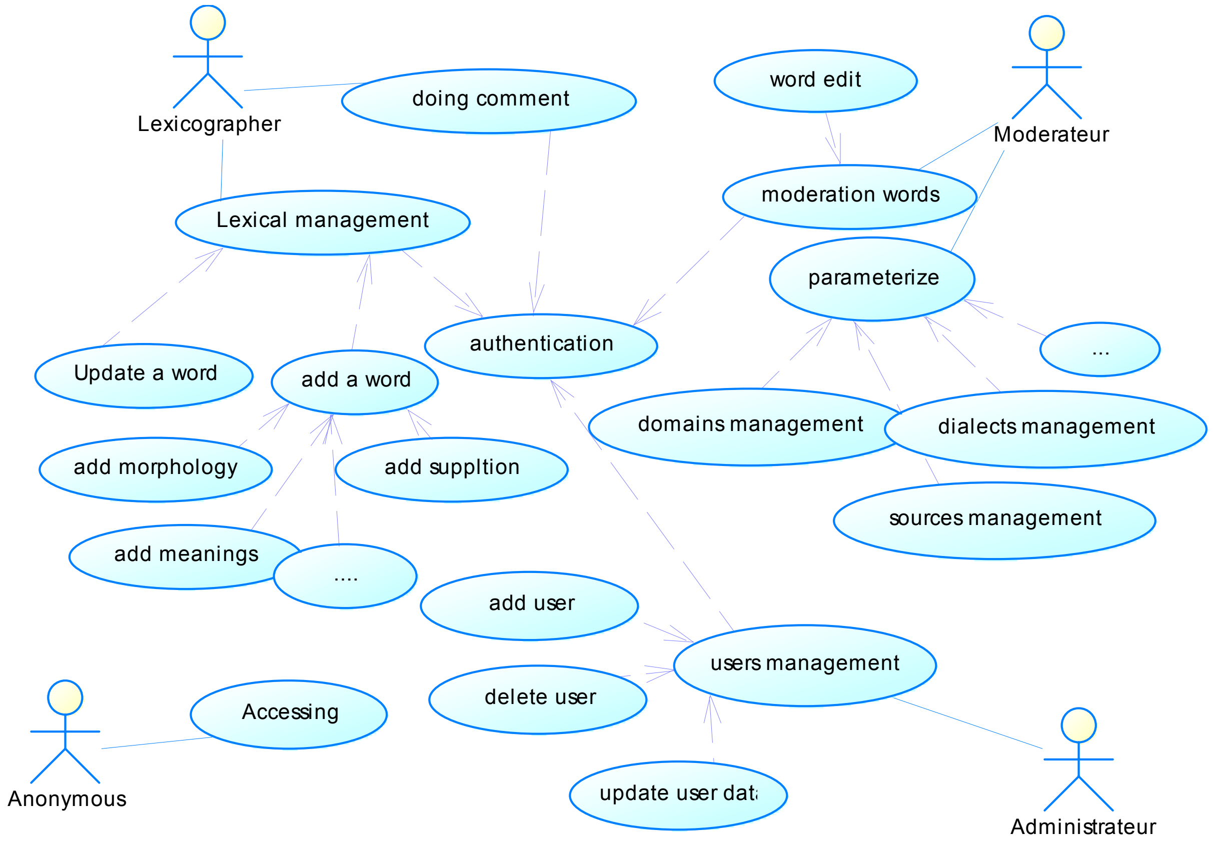


Figure 1: Use cases

4.2 Lexical entry model

The Amazigh data base project is conceived from the beginning in such a way as to provide all the necessary information for each entry. Every Class of the diagram presented in Figure 2 below has its attributes and methods. Each

For example the Lexie Class has for example as attributes: lexie, root, source_lexie, emprunt (borrowing), langue_emprunt (language of borrowing word), neologisme, source_neologisme…etc. As examples of methods used for Lexie Class: ajout_lexie, maj_lexie to update a lexical entry, Rechercher to search a word in the data base. The Lexie Class is the main class of this application it defines a set of meanings (sens Class) and a set of API (International Phonetic Alphabet). It is also associated with other Classes as doublet (free variation), derivee…etc. This Amazigh data base project is conceived from the beginning in such a way as to provide all the necessary information to simplify Amazighe word searching.

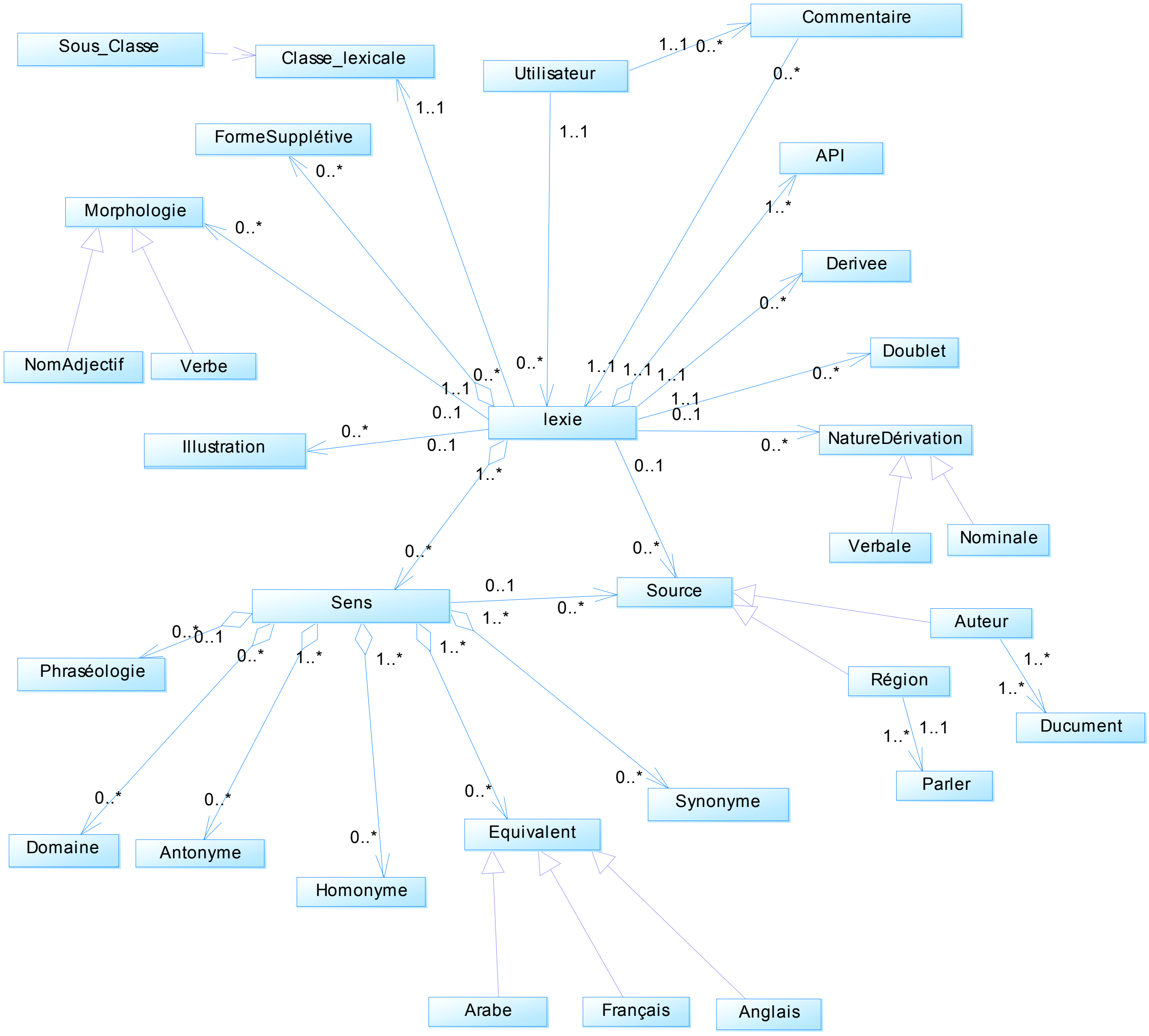


Figure 2: Classes diagram

5. Realization

In order to achieve this application we have used the technology Microsoft dotnet for the implementation and the SQL Server 2000 for the persistence of the classes of this web application, which currently turns on web server Internet Information Server 5.0 which turns on Windows 2000 server.

Références

Ameur, M et al. (2006). Initiation à la langue Amazigh. *Chapitre éléments de morphosyntaxe*, pp. 45--77.

Iazzi, E.M. et al. (2007) Graphie et orthographe de l'Amazigh, IRCAM.

Outahajala, M. (2005) la norme du tri, du clavier et Unicode. In *proceedings of typographie entre les domaines de l'art et l'informatique*, pp. 223--238.

Zenkouar, L. (2004). L'écriture Amazigh tifinaghe et Unicode. In études et documents berbères. n° 22, pp. 175—192.

Building an Arabic Morphological Analyzer as part of an Open Arabic NLP Platform

Lahsen Abouenour[1], Said El Hassani[2], Tawfiq Yazidy[2], Karim Bouzouba[1], Abdelfattah Hamdani[2]

Email: abouenour@yahoo.fr, s.elhassani@iera.ac.ma, elyazidy@iera.ac.ma, karim.bouzoubaa@emi.ac.ma, hamdani@iera.ac.ma

[1] Mohammadia School of Engineers,
Mohamed Vth University-Agdal,
Avenue Ibn Sina B.P. 765 Rabat Morocco

[2] Institute for Studies and Research on
Arabization, B-P 6216, Rabat, Morocco

Abstract

For many reasons (growth of Arabic Internet, greater interest on Arabic Media, etc.), Arabic NLP is facing many challenges. To contribute in answering some of them, we present in this paper an integrated and open Arabic NLP platform. This platform is dedicated to the development of many kinds of Arabic NLP applications. In addition of its openness, this platform is aimed to respect criteria such as standardization, flexibility and reusability. As a first step for the development of the platform, we present also its morphological layer with a focus on Arabic nouns. This analyzer is mainly based on a new classification of the Arabic nouns and provides useful information for other layers (syntax and semantics). Experiments done on selected corpora are very encouraging and the sketched architecture leads to many other interesting future works.

1. Introduction

Nowadays, in the context of NLP in general and Arabic NLP in particular, the following issues can be mentioned as trends: (i) help community researchers to unify their previous and current efforts (in terms of process and data); (ii) encourage the development of open source programs; (iii) find standard protocols and information representation formalisms for a better sharing; (iv) provide conventional languages resources for benchmarking and evaluation; (v) allow the reuse of already programmed modules; (vi) guarantee the portability of programs; etc.

In Arabic NLP many programs (analysers, specific applications such as translators, QA systems, IR systems, etc.) and data (dictionaries, lexicon, and corpora) have been developed. However, most of the time the work is not done as part of an integrated context where openness, standardisation, flexibility, reusability, and so on criteria are respected. Hence, there is a need to contribute in the development of programs and platforms with respect to the current Arabic NLP challenges.

The aim of our group is to follow the trends by developing an integrated Arabic NLP platform with the aim to offer an efficient integrated framework where the above criteria are taken into account. In the context of the present article, we expose the architecture of the platform and detail its morphological module. Indeed, after sketching the architecture of our platform, the first step that has been followed so far was the development of the Arabic morphology layer since it is the basic layer for most of NLP modules.

The structure of the article is as follows. In the second section, we explain the platform architecture with its different layers and modules. Before detailing the morphological module in the fourth section in terms of data and process, we present a brief overview of the Arabic morphology related works in the third section. In section Five, we present our preliminary experiments, and we give interpretation and discussion of the obtained results. Finally, in the last section we draw some conclusions and the future works to be done.

2. The Arabic NLP platform

Our platform is a Java open source multi-layer platform and a modular integrated development environment, dedicated to the development of Arabic NLP applications. As illustrated in Figure 1, the architecture contains the three regular NLP layers (Morphology, Syntax, and Semantics).

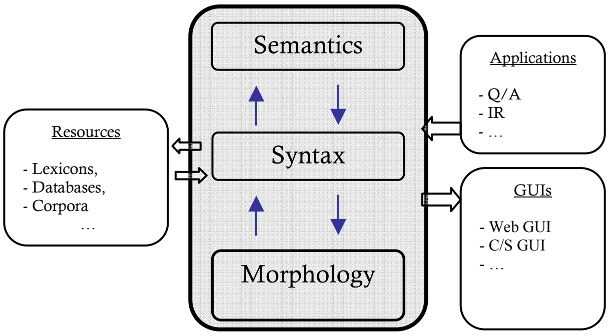
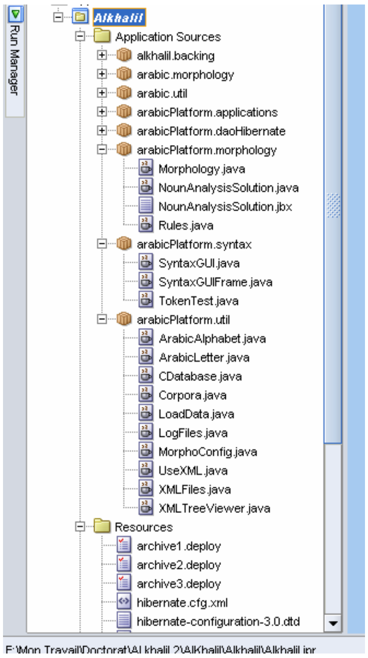


Figure 1: Architecture of the Arabic NLP Platform

Each layer is developed as a reusable Java API (library of classes) – Figure 2. The three layers make use of linguistic resources such as lexicons and corpora. In case the user needs directly to use any one of the layers, it is not necessary to call the libraries but the platform provides also appropriate graphical user interfaces (GUIs) for each layer. On another hand, it is also possible to develop NLP applications (Question/Answering systems, Information Retrieval systems, etc.) by calling one or more of the layers and linguistic resources. For example, it is known that to implement a Q/A system (Benajiba & Rosso, 2007), a light stemmer is enough. Thus, using our platform, the developer of such an application can call a subset of the functions from the morphology layer (`getNounPrefixes`, `getNounSuffixes`, `getNounRoot`, etc.). In addition, the solutions provided by each layer and the flow between them are in XML-like standardized format.



```
Alkhalil
  Application Sources
    alkhalil.backing
    arabic.morphology
    arabic.util
    arabicPlatform.applications
    arabicPlatform.daoHibernate
    arabicPlatform.morphology
      Morphology.java
      NounAnalysisSolution.java
      NounAnalysisSolution.jbx
      Rules.java
    arabicPlatform.syntax
      SyntaxGUI.java
      SyntaxGUIFrame.java
      TokenTest.java
    arabicPlatform.util
      ArabicAlphabet.java
      ArabicLetter.java
      CDatabase.java
      Corpora.java
      LoadData.java
      LogFiles.java
      MorphoConfig.java
      UseXML.java
      XMLFiles.java
      XMLTreeViewer.java
  Resources
    archive1.deploy
    archive2.deploy
    archive3.deploy
    hibernate.cfg.xml
    hibernate-configuration-3.0.dtd
E:\Mon Travail\Doctorat\Al khalil 2\AlKhalil\Alkhalil\Alkhalil.jpr
```

Figure 2: hierarchy of the Arabic NLP Platform APIs

The three layers form a hierarchy. Each layer is built on top of and uses the lower layers. However, a lower layer can be used by itself without the higher layers: the morphology layer (i.e. the associated APIs) can be used directly in any application without the other layers, syntactic layer (i.e. the associated APIs) can be used directly too, etc. Among the goals that have influenced the design of the platform was the goal to achieve a higher level of modularity and independence between its components.

The platform exhibits the following features:

- It guarantees the portability because it is developed with Java
- It could be used and evaluated by the community because it is open source
- It is flexible because the user can use any one of the GUIs or make call of one or more of the libraries
- Thanks to some mapping techniques and to the encapsulation of functions, resources can be downloaded in many proprietary formats (e.g. MySql, Oracle, SQLServer, Access, Ingress, etc.) or in XML standard format; This leads to the possibility of using already built resources and to a better sharing with the use of XML
- The reuse of components; i.e. some API functions can be reused in many contexts. For instance, the function that extracts the root from a word is used in the morphology GUI but could also be used in a Q/A system
- Appropriate documentation is provided to help developers efficiently use the API
- GUIs are provided as web services allowing the community to use the platform from Internet
- Ability to integrate other components

So far, we developed: (i) the structure of the whole architecture. This would allow programmers of the other layers to put their programs at the right place. (ii) the implementation of the morphological layer with its corresponding API and one specific GUI. As a first step we focused on Arabic nouns (verb analysis is to be considered next).

Let us mention also that the other layers are structured as Java interfaces (in term of software engineering), i.e. that a layer could be implemented as a plug in of an already existing tool. This could be the case for example of Buckwalter morphological analyser that can be integrated in our platform with the condition that it respects our API structure.

The analyzer is described in section 4. The next section gives a short presentation of some related works.

3. Related Works

Morphology is the most basic layer over which higher syntactic and semantic layers are built (Attia, 2000). It covers the study of the structure of words and is a term for that branch of linguistics concerned with the forms words take in their different uses and constructions (El-Sadany, 1989).

Morphological processing concerns two different tasks according to the operation type: in generation we produce correct forms using given morphemes, while in analysis we try to identify morphemes for a given word. The number of Arabic morphological analyzers and stemmers is increasing. In this part, we describe some of well known of them (a more detailed list of such analysers could be found in (Al-Sulaiti, 2004)):

- Buckwalter Arabic Morphological Analyzer (2004): Used by Linguistic Data Consortium (LDC) for POS tagging of Arabic texts, this analyzer contains over 77800 stem entries which represent 45000 lexical items. The parser output uses a transliteration system. It has been criticised by (Attia, 2006) for excessive manual processing to state rules and because it analyses only words that appear in Arabic dictionaries.

- Beesley’s Xerox AMA and generator (2001): presented in Java Applet, it’s based on finite state techniques with two level morphological analysis: level for roots and patterns and an other for affixes, prefixes, enclitics and some forms such as prepositions, conjunctions.
- Sakhr’s Arabic Morphological Analyzer[1]: the analyzer of Sakhr Company covers modern and classical Arabic and it identifies the base form by removing all the affixes. It gives also the morphological pattern. However, it suffers from some problems related to disambiguation approach and heterogeneity of processing (Attia, 2000).
- Khoja's stemmer (Khoja and Garside, 1999) attempts to find roots for Arabic words. His stemmer is based on four lists: list of prefixes, list of suffixes, list of roots and list of patterns. When it removes prefixes and suffixes, then it checks a list of patterns to determine whether the remainder could be a known root with a known pattern.
- Larkey's light stemmer (Larkey and Ballesteros, 2002) removes the most frequent suffixes and prefixes (light 10) from the Arabic surface word given. The aim of this light stemmer is not to produce the root but to find the stem.

All of these approaches and others fail to deal with some particular difficulties of Arabic morphology or don’t answer the current Arabic NLP challenges:

- Some of them are not available for the community
- Approaches based on a list of patterns have to do with the fact that some Arabic patterns are assigned to different categories (N, V, A).
- The availability of many types of nominal categories and several rules/processes that are not relevant or adequate to the Arabic morphology.
- The absence of a reasonable solutions for the issue of the fact that some prefixes and suffixes are similar to the basic character (in the beginning and the end) of Arabic words, or the fact that Arabic surface forms are ambiguous unless it can have many vocalized forms.

4. The Arabic Nouns Morphological Analyzer

In this part we describe the Arabic Nouns Morphological Analyzer (فريق المعالجة الآلية للغة العربية , 2007). Its approach is a little bit similar to that of (Buckwalter, 2004) in that it uses a lexicon with explicit linguistic classes. However, our analyzer provides in addition a set of information that (Buckwalter, 2004) do not provide. Moreover, Buckwalter's analyzer is based on about 20 types and misses some prefixes and suffixes (e.g. أ). Our analyzer does not contain any compatibility constraints like buckwalter's, but it uses additional rules like (Habash and Rambow, 2006). So, a lexicon of Arabic nouns has been constructed according to a new categorization. In order to guarantee high performance for our analyzer, we have classified the Arabic nouns into three linguistic classes according to their ability or not to be attached with specific list of suffixes.

Class 1 prohibits the following suffixes: ة، ون، ين. It includes four types of nouns: common nouns, event nouns, proper names and broken plurals.

Class 2 prevents suffixes such as: ان، يْن، ون ين. It contains common nouns (especially kind nouns) and event nouns that can carry the classifier (ة). Note that these two types of nouns are not cited in the first class.

Class 3 forbids one suffix only: ي. It includes derived nouns: present participles, past participles and the exaggerations forms.

We manually developed three dictionaries: one for Arabic nouns, one for prefixes, and one for suffixes. These dictionaries are organized as follows:

- The fields of the nouns dictionary are: the root, the stem, (vocalized and non-vocalized), the category, the type (such as broken plural, proper names, event nouns, etc.), the number and gender. Table 1 is an extraction of this dictionary for the root (ك ت ب).
- For prefixes and prefixes, we have some important morpho-syntactic features for different Arabic morphemes such as number, gender and person in possessive pronouns, or case. We have also some information about its grammatical functions such as coordination, classifier, genitive preposition, a type of the noun, etc.

gender	*number*	*type*	*category*	*Stem vocalized*	*Stem non vocalized*	*root*
						ك ت ب
1	1	2	1	كَتْب	كتب	
2	2	4	1	كُتُب	كتب	
1	1	1	1	كِتاب	كتاب	
1	2	4	1	كُتّاب	كتاب	
1	1	1	1	كُتّاب	كتاب	
2	1	2	1	كِتابَة	كتابة	
1	1	1	1	كُتَيِّب	كتيب	
1	1	1	1	مَكْتَب	مكتب	

Table 1: Extraction from our lexicon of the entry كتب

[1] www.sakhr.com

- Corpus 3 is multi topics article extracted from the ecoworld magazine[4].

	Corpus 1	Corpus 2	Corpus 3
Average solutions per word	2	6	1
Response time (ms)	4300	11329	7857
Performance (%)	87,07	80,53	78,81

Table 1: Results of experiments

In the experiments done we process words (nouns and verbs) in a corpus, then, we calculate performance as:
P=number of words with solutions/number of nouns. Note that we do not verify whether the solution is correct or not, and the number of nouns is calculated manually.

The analyzer reaches an average performance of 82.14 %. The remainder percentage is due to the lack in the dictionary of some proper names (persons, places, etc.) and to the lack of some specific rules that converts a singular noun to its plural (e.g in the lexicon we have سماء and in corpora we can find سماوات).

6. Conclusion and Future Work

We presented in this paper an open platform that answers some of the current and future Arabic NLP challenges such as openness, portability, reusability, and flexibility. After explaining the different components of the platform, we detailed its morphological layer as the module that has been developed so far, supported by some preliminary experiments.

In the next future, we plan to:

- Extend the dictionary with proper names and integrate new morphological rules to improve the performance of the analyzer;
- Confirm the results by further experiments on bigger and conventional corpora (e.g. LDC or Elda corpora);
- Develop the morphological analysis of the Arabic verbs;
- Develop the other modules and layers of the platform;
- Develop some specific applications (e.g. QA systems, IR systems) in the context of the platform.

7. References

Al-Sulaiti, L. (2004). Developing a corpus of Contemporary Arabic. Submitted in accordance with the requirements for the degree of Master of science, The University of Leeds.

Attia, M., (2000). A large-scale computational processor of the Arabic Morphology and applications. thesis submitted to the faculty of engineering, Cairo University.

Attia, M. (2006). An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks. The Challenge of Arabic for NLP/MT Conference, the British Computer Society, London.

Beesley, K., (2001). Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. Proceedings of the Arabic Language Processing: Status and Prospect, 39th Annual Meeting of the Association for Computational Linguistics. Toulouse, France.

Benajiba, Y., Rosso, P. (2007). Towards a measure for Arabic corpora quality. In Proceedings of the seventh International Conference CITALA 07. Rabat, Morocco. pp. 213--220.

Buckwalter, T. (2004). Issues in Arabic Orthography and Morphology Analysis. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, Geneva.

El-Sadany, T. A., Hashish, M. A. (1989). An Arabic Morphological System. IBM SYSTEM JOURNAL vol 28-no 4.

Habash, N., Rambow, O. (2006). Morphological Analysis for Arabic Dialects. In Proceedings of COLING-ACL, Sydney, Australia.

Khoja, S., Garside, R. 1999. Stemming Arabic text. Computing Department, Lancaster University, Lancaster.http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps.

Larkey, L. S., Ballesteros, L., 2002. 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.

فريق المعالجة الآلية للغة العربية، 2007 .التحليل الصرفي للأسماء العربية .ندوة دولية حول المعالجة الآلية للغة العربية، معهد الدراسات و الأبحاث للتعريب، الرباط، المغرب.

[4] http://www.ecoworldmag.com

The goal of our analyzer is not only to remove all prefixes and suffixes from the surface Arabic words to find roots or stems, but also to:

- introduce many special information about stems and its prefixes and suffixes
- find all possible results existing in the Arabic lexicon.

Finally, there are a set of processes (rules) that apply in particular cases. For example, we have to predict that the form of ء (hamza) can be ـئ or ؤ, the omission of ة when we move from singular forms to plurals ones in words like (معلمة-معلمات), and so on.

Figure 3 below describes how the morphological analyzer processes.

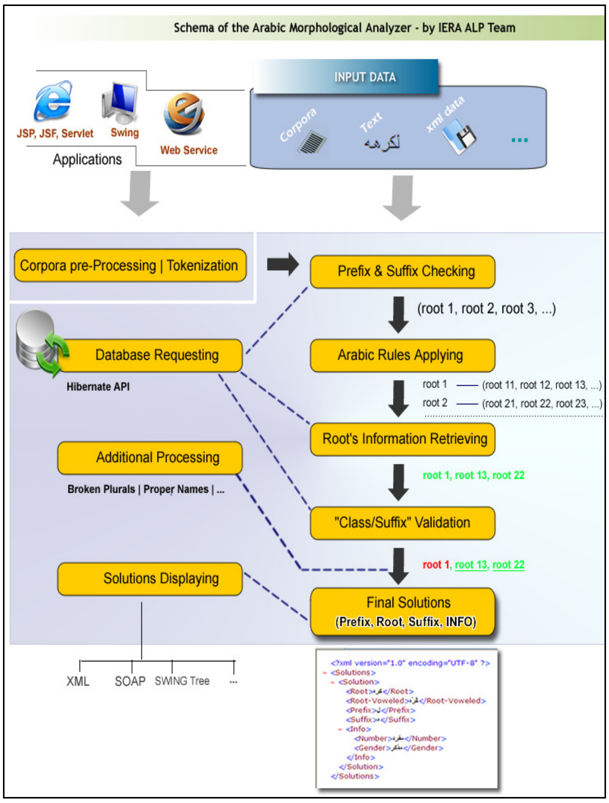


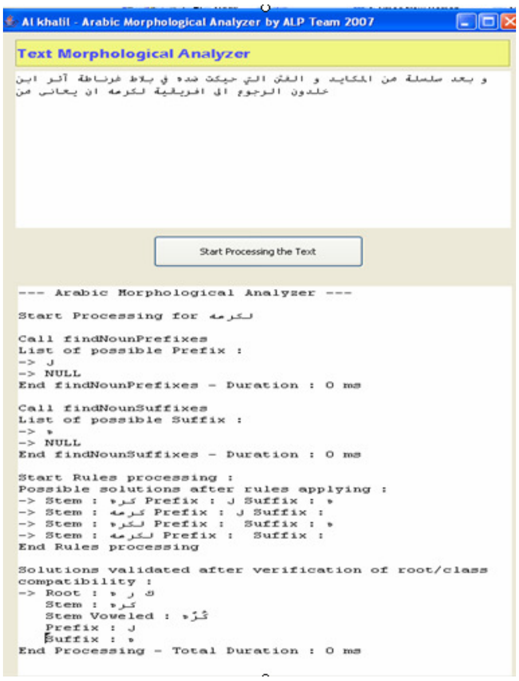
Figure 3: Schema of our Arabic Nouns Analyzer

Given an Arabic word, text or corpora, our analyzer begins by pre-processing it applying text segmentation. Next, for each word, it tries to look up a list of possible solutions. They are composed of a prefix, a root and a suffix. For each possible solution we apply a set of specific Arabic rules according to its elements (e.g. if prefix is ل then apply rule 1). After this step new possible solutions are generated. All those possible solutions are checked from the lexical database. After the category validation, the analyzer returns the final solutions under different formats (XML, GUI, etc.).

Additional processing can be done before displaying final solutions in order to resolve some particular problems like broken plurals and proper names, or to disambiguate with vocalization.

Figure 4 below shows a snapshot of one GUI using our analyzer API:

Figure 4 below show a snapshot of two GUI using our analyzer API:


Al khalil - Arabic Morphological Analyzer by ALP Team 2007
Text Morphological Analyzer
Start Processing the Text

و بعد سلسلة من المكايد و الفتن التي حيكت ضده في بلاط غرناطة آثر ابن خلدون الرجوع الى افريقية لكرمه ان يعاني من

```
--- Arabic Morphological Analyzer ---

Start Processing for لكرمه

Call findNounPrefixes
List of possible Prefix :
-> ل
-> NULL
End findNounPrefixes - Duration : 0 ms

Call findNounSuffixes
List of possible Suffix :
-> ه
-> NULL
End findNounSuffixes - Duration : 0 ms

Start Rules processing :
Possible solutions after rules applying :
-> Stem : كره Prefix : ل Suffix : ه
-> Stem : كرمه Prefix : ل Suffix :
-> Stem : لكره Prefix :  Suffix : ه
-> Stem : لكرمه Prefix :  Suffix :
End Rules processing

Solutions validated after verification of root/class
compatibility :
-> Root : ك ر ه
   Stem : كره
   Stem Voweled : كُرْه
   Prefix : ل
   Suffix : ه
End Processing - Total Duration : 0 ms
```

Figure 4: GUI for analyzing a given Arabic text

In the first figure we introduce an Arabic text to be processed. The second figure shows an example of our analyzer processing report where indicated the results of each step (getting prefixes, getting suffixes, applying rules, etc.). The analysis of the word لكرمه reports two possible prefixes (ل, NULL) and two suffixes (ه,NULL). After applying linguistic rules to the possible solutions above, the system validate the only solution given as result (root : ك-ر-ه stem : كره prefix : ل suffix : ه).

As mentioned before, the solutions are also stored under XML format.

5. Results and Evaluation

The experiments were done with a 1.66 GHZ core 2 duo processor with 1 Go RAM and 1 Mo cache. Table 1 provides the detailed results of the three corpora used for the evaluation:

- Corpus 1 is a culture content extracted from a cultural web site[2].
- Corpus 2 is a one topic article extracted from el akhbar online newspaper[3].

[2] http://fatiha.sosblog.fr/Premier-blog-b1/Day7-b1-m20080207.htm

[3] http://www.elakhbar.org.eg/akhbarelyom/Issues/

Morpho-syntactic tagging system for Arabic texts

A. Yousfi , A. El jihad, and L. Aouragh,
IERA, Allal Al-Fassi,B.P :6216, Rabat-Institutes.Morocco.Tel 037773012, Fax 037772065
Yousfi240ma@yahoo.fr, jihad@iera.ac.ma, Aouragh@hotmail.com

Abstract

Text tagging is a very important tool for various applications in natural language processing, namely the morphological and syntactic analysis of texts, indexation and information retrieval, "vocalization" of Arabic texts, and probabilistic language model (n-class model).
However, these systems based on the lexemes of limited size, are unable to treat unknown words consequently.
To overcome this problem, we developed in this paper, a new system based on the patterns of unknown words and the Hidden Markov Model (HMM).
The experiments are carried out in the set of labeled texts, the set of 3800 patterns, and the 52 tags of morpho-syntactic nature, to estimate the parameters of the new model HMM.

Index Terms— Hidden Markov Model, Morpho-syntactic tagging, Arabic text, , pattern.

1. Introduction

In this paper, we have developed a new system based on the patterns of unknown words and the Hidden Markov Model (HMM). The originality of this work is to overcome the deficiency observed in other systems that do not treat these unknown words. The experiments are carried using tagged texts, a set of 3800 patterns, and 52 tags of morpho-syntactic nature, to estimate the parameters of the new HMM model.
The automatic tagging of texts is a process that consists of assigning morphological, syntactic, semantic, prosodic, critical, morpho-syntactic information to segment of texts (generally a word) with (Veronis, 2000), (Vergne, 1998).
It consists of three steps (Thi Minh &all, 2003), (Paroubek & all):
1) Segmentation of the text into lexemes.
2) Tagging that assigns for each identified lexeme the whole of the possible morpho-syntactic tags.
3) Disambiguation: in trivial cases there will be only one tag per word.
Generally, there are two main approaches to part-of-speech tagging: rule-based tagging and probabilistic.
Among the difficulties which arise in the systems of tagging are the unknown words.
In this paper we developed an approach to solve the problem of the unknown words by using the notion of word patterns. This approach is integrated into the morpho-syntactic system of tagging based on the model of hidden Markov, developed at the IERA (EL JIHAD & Yousfi, 2005).

2. Arabic word patterns

The Arabic language has a particular characteristic, namely verbs and derived nouns can be classified into patterns. The construction of a word pattern is given by the procedure described in (Al Ghalayni, 2000) which extracts the radical letters composing its root from the word, then replaced as follows: the first radical letter is replaced by "ف", the second letter by "ع", the third by "ل" and the fourth by "ل".
Example:
- The patterns word of "تَدَحْرَجَ" is "تَفَعْلَلَ".
- The patterns word of "كَتَبْتُ" is "فَعَلْتُ ".

In Arabic, every word has many patterns. For the research of the pattern word, we developed a measure D, which measures the degree of similarity between the word and each pattern.

$F = \{f_1, f_2, ..., f_N\}$ is the set of all patterns and w is a word. The set of patterns of w is given by :

$$F(m) = \{f_i \ / D(m, f_i) \geq 0\}$$

3. Tagging by probabilistic method

The choice of the most probable tag of a word is made in comparison with the history of the last tags which have been just affected. This history is limited to one or two previous tags. The probabilistic method supposes that one has a sufficient training corpus to allow a reliable estimation of the probabilities (Habert & all, 1997).
Let:
ph = $w_1...w_p$ a sentence of words : w1,...,wp, in the vocabulary V.
E= $\{et_1,...,et_N\}$ the set of morpho-syntax tags.
The morpho-syntactic tagging of the ph by tags in E which is based on the probabilistic approach consist to find a set of tags $et_1^*,...,et_p^*$ associated with the sentence ph, such as :

$$et_1^*,...,et_p^* = \arg\max_{et_1,...,et_p} \Pr(w_1,...,w_p,et_1,...,et_p) \quad (1)$$

The problem which arises in this formulas is words that do not exist in V.

To solve the equation (1) by tagging into account of this problem, we adapted the Hidden Markov Model by introducing the patterns notion of unknown words.

3.1 morpho-syntactic tagging by using word PATTERNS

The HMM model of order 1 by considering the word patterns, is a process with:

- Xt is a Markov chain of order 1, with value is in a finished set of states *Q={q₁,...,q_N}*, X_t checks:

  $$\Pr\,(X_{t+1} = q_j \,/\, X_1 = q_1,...,\ X_t = q_i) =$$
  $$= \Pr\,(X_{t+1} = q_j \,/\, X_t = q_i) = a_{ij}$$
  $$\Pr(X_1 = q_i) = \pi_i \quad i = 1,...,N$$

  - a_{ij} is the transition probability between q_i and q_j
  - π_i is the probability that q_i is an initial state.

- Y_t is an observable process with values in a measurable unit *Y*, Y_t checks:

  $$\Pr\,(Y_t = y_t \,/\, X_1 = q_1,..., X_t = q_i, Y_1 = y_1,..., Y_{t-1} = y_{t-1}) =$$
  $$= \Pr\,(Y_t = y_t \,/\, X_t = q_i) = b_i(y_t) = b_{it}$$

  - b_{it} is the emission probability of the observation y_t from state q_i.

- Z_t is an observable process with values in a measurable set *Z* . Z_t checks:

  $$\Pr(Z_t = z_t \,/\, X_1 = q_1,..., X_t = q_i, Z_1 = z_1,..., Z_{t-1} = z_{t-1}) =$$
  $$= \Pr(Z_t = z_t \,/\, X_t = q_i) = d_i(z_t) = d_{it}$$

  - d_{it} is the emission probability of the observation z_t starting from state q_i .

In the continuation one, we will suppose that the process: *(X, $Y_{t\,t}$, Z_t,)* is HMM of order 1:

- $X_t = et_{it}$ representing the morpho-syntactic tag, with value is in E.
- $Y_t = w_t$ representing the words of the vocabulary *V = {w₁,...,w_L}*,
- $Z_t = f_t$ representing the patterns words.

Note:

This model is defined by a parameter vector λ = *(Π,A,B,D)*:

- $\Pi = \{\pi_1,...,\pi_N\}$ The set of the initial probability.
- *A = $(a_{ij})_{1\leq i,j\leq N}$* : the matrix of the transition probabilities.
- *B = $(b_{it})_{1\leq i\leq N\ et\ 1\leq t\leq L}$* : the matrix of the emission probabilities of the words from the states.
- *D = $(d_{it})_{1\leq i\leq N\ et\ 1\leq t\leq L}$* : the matrix of the emission probabilities of the patterns words from the states.

3.2 LEARNING PROCEDURE (PARAMETERS OF ESTIMATION)

Training is a necessary operation for a tagging system which makes to estimate the parameters of the model λ = (Π,A,B,D). An incorrect or insufficient training decreases the performance of the tagging system.

In general there are three methods to estimate these parameters (Yousfi, 2001):

- Estimation by the Likelihood Maximum which is carried out by the algorithm of Baum-Welch (Baum, 1972) or Viterbi algorithm (Celux & Clairambault, 1992).
- Estimation by the maximum a posteriori (John).
- Estimation by a mutual information maximum (Kapadia & all, 1993).

In our case, we used the estimation by the Likelihood Maximum.

If we take the training set R = {ph1,...,phk} composed by the tagged sentences ph1,...,phk.. The Formulas of estimate the parameters of λ = (Π,A,B,D), are given by:

$$a_{ij} = \frac{\sum_{n=1}^{k} \text{number } \textit{of} \text{ transition } et_i\ et_j \textit{ in the sentence } ph_n}{\sum_{n=1}^{k} \text{number of state } et_i \textit{ in } ph_n}$$

$$\pi_i = \frac{\sum_{n=1}^{k} \delta\,[et_i \text{ is initial state}]}{k}$$

$$b_{it} = \frac{\sum_{n=1}^{k} \textit{number of word } w_t \textit{ are the tag } et_i \textit{ in } ph_n}{\sum_{n=1}^{k} \text{number of state } et_i \textit{ in } ph_n}$$

$$d_{it}\ \frac{\sum_{n=1}^{k} \textit{number of the pattern } f_t \textit{ are the tag } et_i \textit{ in } ph_n}{\sum_{n=1}^{k} n\text{uumber of state } et_i \textit{ in } ph_n}$$

With:

$$\delta[x] = \begin{cases} 1 & \textit{if the evenement} \quad x \textit{ is true} \\ 0 & \textit{else} \end{cases}$$

3.3 AUTOMATIC TAGGING BY THE ADAPTED VITERBI ALGORITHME

For a faster calculation of the equation, we have adapted the Viterbi algorithm (Fornay, 1973) to solve this equation (1).

We note by:

$$\delta_t(et_j) = \max_{et_{i_1}...et_{i_t}} \Pr(\text{w}_1 ... \text{w}_t, et_{i_1} ... et_{i_t}) \quad (2)$$

$$\textit{with} \quad et_{i_t} = et_j$$

To solve the problem of the unknown words, we have introduced the process of the patterns of these words into the formula (2). This formula becomes (Yousfi, 2001).:

$$\delta_t(et_j) = \begin{cases} \max\limits_{\text{et}_\text{i}} \delta_{t-1}(et_i).a_{ij}.b_j(w_t) & si \quad w_t \in V \\ \max\limits_{\text{et}_\text{i},\, f_k \in F(w_t)} \delta_{t-1}(et_i).a_{ij}.d_j(f_k) & \text{sin}\,\textit{on} \end{cases}$$

$F(w_t)$: The set of all possible patterns of w_t .

We calculate this formula for all the values t = 1...,T and j=1...,N.

At the end, the optimal path is obtained by using a recursive calculation of this formula.

4. Experimental results

The Experimental work was completed in four major steps:

- Step of definition of the set of tag and construction of the corpus of training. The definition of our own morpho-syntactic set of tag was particularly delicate; this phase was carried out in collaboration with linguists to satisfy the need for the projects under development at IERA. This set of tag consists of 52 tags of morpho-syntactic nature which are illustrated in Table1:

Signification	Tags
Prefix	س.ب
Active participle	إ.ف
Genetive particle	ح.ج
Exception particle	ح.إ
Proper name	إ.ع
Deiclic perticle	إ.إ
Exaggeration noun	إ.غ

Table1 : Exaples of some morpho-syntactic tag used in our system.

The training corpus is composed of a whole of sentences representing the major morphological and syntactic rules used in Arabic. This corpus was tagged manually by linguists.

أَكْرَمْتُ/ف.م.م.م ال/س.ب مُجْتَهِدِينَ/إ.ف ./.
أَحْسَنْتُ/ف.م.م.م إِلَى/ح.ج ال/س.ب مُجْتَهِدِينَ/إ.ف ./.
جَاءَ/ف.م.م.م ال/س.ب مُسَافِرُونَ/إ.ف إِلَّا/ح.إ سَعِيدًا/إ.ع ./.
قَلَّمَا/ف.ج فَعَلْتُ/ف.م.م.م هَذَا/إ.إ ./.

Example of an extract from our learning corpus

- Step of construction of the data base of the forms of the words. This base consists of 3800 patterns, and is generated by a morphological generator of verbs developed at the IERA. This base has been enriched by the patterns of derived names (table 2).

- Step of estimate the parameters of adapted hidden Markov model.
- Automatic step of tagging and reestimation of the parameters of hidden Markov model. To carry out these two last steps, we have developed an application in C programming language which is based on three modules:

i) Module of determination of the form of a given word.
ii) Module which makes it possible to carry out the phase of training,
iii) Automatic module of tagging of rough corpus. The latter is corrected manually to be used for a reappraisal of the parameters of the model of adapted hidden Markov. The programs are evaluated on the basis of version of text. Results the error rate is measured on a whole of test containing 500 sentences and it is shown in Table3.

فاعل	إ.ف
متفاعل	إ.ف
إفعال	م.ص
تفعيل	م.ص
مفعول	إ.م
مفعل	إ.م
فعلت	ف.م.م.م
أفعل	ف.ض.م.م

Table 2 : Example of an extract from the database patterns words

	Ensemble test
old system (HMM)	3%

Table 3 : The error rate on the entire test to the old system

The error rate is measured on a test set containing 500 sentences. The latter are a sequence of words involved in the old system vocabulary (EL JIHAD & Yousfi, 2005).

2% of these errors are the result of a bad assignment of tags by the system.

1% comes from problems of unpresented transition between tags within training set.

	Ensemble test
new system (adapted HMM)	1.78%

Table 4 : The error rate on the entire test for the new system

For the new system, the error rate is measured on a test set consisting of 500 sentences containing unknown word.

0.92% of these errors come from problems of unpresented transitions between tags in the training set. The remaining of mistakes is the result of the incorrect tagging.

5. Conclusion And perspectives

By analyzing the results, we have note that the introduction of the concept of the patterns words has decreased the effect of the problem of unknown words. Moreover this new system has successfully labelled 1.78% sentences containing these unknown words.

As for the perspectives of this work, we intend to introduce syntactic rules in our system to address the problem of transitions.

REFERENCES

Veronis, J. (2000). *Annotation automatique de corpus: panorama et état de la technique*. Ingénierie des langues. Paris, HERMES Sciences Europe. pp.111-128.

Vergne,J. et Emmanue.l G. (1998). *Regards théoriques sur le Tagging*. (TALN1998), Paris, France.

Thi Minh, Hu., Laurent, R. et Xuan L. (2003). *Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens*. (TALN2003), Batz-sur Mer.

Paroubek, P. et Martin, R. Etiquetage morpho-syntaxique. Ingénierie des langues. Paris, HERMES Sciences Europe. pp.131-150.

EL JIHAD A.et Yousfi A. (2005). *Etiquetage morpho-syntaxique des textes arabes par modèle de Markov caché*. (RECITAL 2005)., 06-10 Juin 2005 Dourdan, France.

Al Ghalayni, M. (2000). المكتبة العصرية. جامع الدروس العربية

Habert, B., A, Nazarenko, et A, Salem. (1997). Les linguistiques de corpus, Armand colin / Masson.Paris.

Yousfi, A. (2001). Introduction de la Vitesse d'élocution dans un modèle de reconnaissance automatique de la parole. Thèse de doctorat, université : Mohamed premier, Oujda, Maroc. 115p.

Baum,L. (1972). *An inequality and association maximization technique in statistical estimation for probabilistic functions of Markov processes*. Inequality, vol. 3.

Celux, G.et J. Clairambault. (1992). *Estimation de chaînes de Markov cachées: méthodes et problèmes*. Journées mathématiques CNRS sur les approches markoviennes en signal et images.

John. R. Mathematical Statistics and data analysis. pp 511-540.

Kapadia. S, V. Valtchev et S.J. Young. (1993). *MMI training for continuous phoneme recognition on the TIMIT database*. Proc.ICASSP, pp. II.491-494, Minneapolis.

Fornay,D. R. (1973). *The Viterbi Algorithm.* Proc. IEEE, vol. 61, n° 3.

Guidelines for Annotation of Arabic Dialectness

Nizar Habash, Owen Rambow, Mona Diab and Reem Kanjawi-Faraj

Center for Computational Learning Systems
Columbia University
New York, NY, USA
{habash,rambow,diab}@ccls.columbia.edu

Abstract

The Arabic language is a collection of variants with phonological, morphological, lexical, and syntactic differences comparable to those among the Romance languages. However, throughout the Arab world, the standard written language is the same, Modern Standard Arabic (MSA). MSA is based on Classical Arabic and is itself not a native spoken language. MSA is used in some official spoken communication, such as newscasts, parliamentary debates, etc. Other forms of Arabic (generally referred to as "dialects" of MSA) are what people use for daily spoken communication. In this paper, we focus on the issue of creating standard annotation guidelines identifying dialect switching between MSA and at least one dialect in written text. These guidelines can form the basis for the annotation of large collections of data that will be used for training and testing automatic approaches to dialect identification and automatic processing of Arabic which exhibits dialect switching. We report on some initial annotation experiments: we discuss statistical distributions of labels in a small corpus (~19K words) that we annotated according to the guidelines and present inter-annotator agreement results.

1. Introduction

The Arabic language is a collection of variants with phonological, morphological, lexical, and syntactic differences comparable to those among the Romance languages (see for example نخله 1959, Butros, 1973, Omar 1976, السامرائي 1981, Brustad 2000, Bateson 2003, Holes 2004; for computationally oriented summaries of the linguistic situation, see Habash 2006, Diab and Habash 2007). However, throughout the Arab world, the standard written language is the same, Modern Standard Arabic (MSA, فصحى *fuSHaý*),[1] also used in some official spoken communication (newscasts, parliamentary debates, etc.). MSA is based on Classical Arabic and is itself *not* a native spoken language. Other forms of Arabic (generally referred to as "dialects" of Arabic) are what people use for daily spoken communication. In unofficial written communication, in particular in the now growing electronic media, often ad-hoc transcriptions of dialects are used. Arabic dialects are usually divided into four geographic groups: Gulf (including all dialects of the Arabian Peninsula and Iraqi), Levantine (including Syrian, Lebanese, Palestinian), Egyptian (including Libyan and Sudanese) and Maghreb (covering all dialects found West of Libya). Other factors such as the bedouin/sedentary distinction and the distinction between rural (village) and urban communities also define some sub-dialectal variations.

MSA and the dialects thus form a prototypical case of "diglossia" (Ferguson 1959). In a diglossic situation, a "high" language is used in public or prestigious communicative situations (media, government) and is written, while a "low" language is used in private communicative situations (family, daily life) and is usually not written. The two forms co-exist; all speakers master the low form, and most speakers also master the high language to a greater or lesser extent. Arabic conforms to the original definition of diglossia given by Ferguson (1959) in that the two forms are closely related.

In diglossic linguistic situations, one frequently finds cases of code switching, i.e., the use of one or more languages, language variants, or dialects in one discourse, and often within one sentence. We will use the term "dialect switching" in this paper to mean the use of two or more variants of Arabic in one discourse. Dialect switching is well attested in Arabic; see for example (Badawi and Hinds 1986) for a proposal to divide Arabic into five levels, each characterized by the extent of the contributions from MSA (or classical Arabic), from dialectal Arabic, and from foreign languages. Different levels correspond to different sociolinguistic parameters, including education level of the discourse participants, as well as discourse purpose and setting. We adopt the assumption that different types of dialect switching happen in different discourses.

In this paper we focus on the issue of creating standard annotation guidelines identifying dialect switching between MSA and at least one dialect in written text (including transcripts of spoken Arabic). These guidelines can then be used to annotate large collections of data that will be used for training and testing natural language

[1] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007). This scheme extends Buckwalter's transliteration scheme (Buckwalter, 2004) to increase its readability while maintaining the 1-to-1 correspondence with Arabic orthography as represented in standard encodings of Arabic, i.e., Unicode, CP-1256, etc. The following are the only differences from Buckwalter's scheme (indicated in parentheses): Ā آ (|), Â أ (>), ŵ ؤ (&), Ǎ إ (<), ŷ ئ (}), ħ ة (p), θ ث (v), ð ذ (*), š ش ($), Ď ظ (Z), ς ع (E), γ غ (g), ý ى (Y), ã ـً (F), ũ ـٌ (N), ĩ ـٍ (K).

processing (NLP) tools which can identify when dialect switching occurs in a document, and which dialects are involved. This is important, as dialect switching is quite common – as we discuss in Section 3.1, even in edited newswire (which should be all MSA), we find 3.8% of segments to have dialectal influence, and this percentage goes up to 72.3% for broadcast conversations on television (in which participants are usually expected to communicate in MSA as well). The problem is that most existing NLP tools have been developed for pure MSA, for example morphological taggers (Habash and Rambow 2005, Diab et al 2007) or parsers (Bikel 2002, Maamouri et al. 2004a, Maamouri et al. 2006). To have these tools handle Arabic as it is actually produced (namely, with dialect switching), we need to be able not only to handle pure MSA and pure dialectal data,[2] but we also need to develop computational models that can detect and incorporate dialect switching. The proposed annotation guidelines will provide a first step towards creating such computational models, and, generally, towards a broad empirical study of dialect switching in Arabic.

The task of defining annotation guidelines for Arabic dialect switching (or in fact any dialect switching) is complex, because the boundaries between MSA and a dialect are not well defined. If we are studying code switching between, for example, Arabic and English, we can almost always determine from which language a word, a morpheme, even the pronunciation is taken. The only major methodological difficulty is distinguishing borrowings (words that have entered the general vocabulary of language A from language B, such as *algebra* or *oud* in English) from nonce borrowings (words from language B used spontaneously by a speaker while speaking predominantly in language A). In the case of dialect switching, we cannot readily identify borrowings (whether nonce borrowings or regular borrowings) at all. Especially complicating is the fact that Arabic orthography often omits all short vowel diacritics, which may otherwise distinguish between different dialects.

In the rest of this paper we present and exemplify our guidelines (Section 2) and discuss a preliminary study annotating a small corpus with these guidelines and computing inter-annotator agreement (Section 3).

2. Dialect Switching Annotation

When investigating dialect switching (and code switching in general), we can distinguish several (potentially) orthogonal levels of annotation: phonology, morphology, lexicon, and syntax. Since we are interested in written language (including transcribed speech) for the sake of this paper, we omit the phonological level and replace it by orthography. One approach to annotation would be to annotate each of the levels separately. However, in this paper, we group the annotation decisions into two decisions for the word and the segment level (plus one additional annotation decision for the document level). The decisions at the word and segment level are judgments on a precisely defined scale, which we refer to as the "dialectness" scale.

Given the difficulty of determining whether a word (or a morpheme) is a borrowing at all in a dialect code switching situation, we have decided to *always* code a word (or morpheme) which can be construed in context as being an MSA word (or morpheme) as MSA. Put differently, our default assumption is that the text is MSA, and we are annotating only clear evidence of dialect influence.

Our approach – two different judgments at the word and segment levels – represents a calculated shortcut, in that it does not provide detailed information at the orthographic, morphological, lexical, and syntactic levels. Instead, based on a preliminary study, we make the assumption that certain combinations are so rare that a simplified annotation scheme is warranted. We acknowledge that a more complex annotation scheme may be required for certain types of studies or computational needs.

2.1 Word-Level Dialect Annotation

Decisions at the word level reflect on orthography, morphology and the lexicon. We annotate the dialectness of each word as a choice among four levels. To determine the correct level, we must first determine the *lexeme* and the *inflectional morphemes* of the word. The two notions are closely related. The lexeme is an abstraction over all possible inflectional morphemes. For example, كتاب *kitAb* 'book', الكتابان *AlkitAbAn* 'the two books', وبكتبكم *wabikutubikum* 'and with your books' all are variants of the same lexeme, which, like all nouns, is usually referred to with the citation form of the singular masculine, كتاب *kitAb* 'book'. Although the word-level annotation task is concerned with the 'words', the context is still relevant to some degree. The annotators need to determine what the meaning of the complete segment is first before proceeding to annotate the different words. This is important since certain words may potentially mean different things in different contexts and deserve different treatments. For example, the word بايع *bAyç* in بايع سيارتك انت؟ *bAyiç say~Artak inta?* 'are you selling your car?' clearly means 'selling' (*bAyiç*) not 'offered allegiance' (*bAyaç*). The first reading is clearly dialectal (from بائع *bAŷiç*) whereas the second reading is MSA. Thus, in this context, this word would be coded as being a dialectal lexeme (since there is no plausible MSA reading).

The following are the four levels for word dialectness annotation:

Word Level 0 is used for *pure MSA* words. These words are standard MSA lexemes with the contextually appropriate MSA inflectional morphology; they have standard MSA orthography with no typographical errors, e.g., يكتبون *yaktubuwn* 'they write', المساجد *AlmasAjid* 'mosques', سيقوم *sayaquwm* 'he will rise', اعيادكم

[2] For example, see (Habash and Rambow 2006) and (Chiang et al. 2006) for NLP tools for the dialects.

AçyAdukum ‘your holidays’, بغداد *baγdAd* ‘Baghdad’, سيليكون *siyliykawn* ‘silicon’, and واشنطن *wAšinTuwn* ‘Washington’.

Word Level 1 refers to *MSA with non-standard orthography*. These words are standard MSA lexemes with the contextually appropriate MSA inflectional morphology, but something is strange in the spelling: either a non-standard spelling possibly inspired by non-standard pronunciation, regional variation in spelling, or typographical errors. We do not ask the annotators to choose one of these options as they are often hard to distinguish. The annotators also do not indicate the correct spelling. Examples include مساجذ *masAjið* (instead of مساجد *masAjid* ‘mosques’, presumably a spelling error), فسطان *fusTAn* (instead of فستان *fustAn* ‘dress’, presumably dialectal spelling) and هدا *hadA* (instead of هذا *haðA* ‘this’ dialectal spelling or spelling error).

Word Level 2 indicates an *MSA word with dialect morphology*. These words are standard MSA lexemes but they have at least one morpheme which is clearly a dialectal morpheme (from any of the dialects). A very common example of this is the use of the +ب *b*+ prefix in conjunction with an otherwise entirely acceptable MSA verb, for example بيذهب *biyaðhab* ‘he goes’. Note that the +ب *b*+ prefix can appear with nouns in MSA but not with verbs. A common spelling of the Levantine/Egyptian verb ‘I write’ is بكتب *baktub*; although this word looks like the MSA *bikutub* ‘in books’, the syntactic and semantic contexts are used to disambiguate. Examples of other dialectal morphemes often found in conjunction with MSA words include +ح *Ha*+ (Egyptian/Levantine ‘future tense’), +ع *ça*+ (Levantine preposition ‘on/to’), +د *da*+ (Iraqi ‘present tense’), +ك *ka*+ (Moroccan ‘present tense’), +غ *γa*+ (Moroccan ‘future tense’), م++ش *ma*++*š* (Egyptian/ Levantine negation circumfix) and +ش *š*+ (Iraqi question marker).

Word Level 3 indicates a *dialect lexeme.* These words are words which clearly would never be used in written or spoken MSA by an educated speaker/writer. This level does not include orthographic variants of MSA words, these are coded as Word Level 1. Prime examples are the negation marker مش *miš* ‘no/not’ and its variants and other dialectal morphemes that can be spelled separately from the word: ح *Ha* (Egyptian/Levantine ‘future tense’), ع *ça* (Levantine preposition ‘on/to’), and عم *çam* (Levantine verbal particle marking progressiveness), etc. Other examples of purely dialectal lexemes include: بزونة *baz~unaħ* (Iraqi for ‘cat’). Dialectal lexemes that have MSA homophones are also marked if the context shows that the word is clearly dialectal, for example عافية *çAfyaħ* (Moroccan for ‘fire’ but MSA/Levantine for ‘health’). Note that at this level, orthography is irrelevant, as there is no standard orthography for the dialects anyway. We also do not consider morphology: a dialect word with MSA-only morphology is also coded as Word Level 3.

2.2 Segment-Level Dialect Annotation

Beyond word annotation, the annotators are asked to judge the whole segment (sentence/utterance) all at once in terms of the quality of the MSA. This annotation is necessary to address cases that at the word level seem all MSA, but it is clear that the sentence is dialectal or mixed because of lexical issues involving multi-word lexemes, or because of syntax. The judgment is on a scale from 0 to 4.

Segment Level 0 is defined to be *perfect MSA***.**

Segment Level 1 is *imperfect MSA*. Here, the source is trying to produce MSA, but some dialectal phenomena are sneaking in (dialectally inspired orthography revealing pronunciation, some dialectal morphology, incorrect case or mood, incorrect subject-verb agreement, or perhaps an isolated dialectal lexeme). A segment cannot be Segment Level 1 if there are more than minimal number words of Word Level 3 – in practice, we have yet to define this threshold.

Segment Level 2 is *Arabic with full dialect switching***.** It is not clear whether the writer is aiming for writing in MSA, or in dialect. A segment cannot be Segment Level 2 if all words are Word Level 2 or Word Level 3.

Segment Level 3 refers to *dialect with MSA incursions*. The source is producing dialectal Arabic, but uses some clichés or words clearly borrowed from MSA. A segment cannot be Segment Level 3 if all words are Word Level 3.

Segment Level 4 is used to mark *pure Dialect.* Here, the writer is producing pure dialectal Arabic. A segment of Segment Level 4 has, in general, at least one word of Word Level 2 or Word Level 3.

2.3 Source-Level Annotation: Home Dialect of Speaker/Writer

After the previous two annotations are done, the annotator may have a reasonably good idea of the home dialect of the speaker/writer. This guess will be marked. The annotators are encouraged to use any knowledge they can muster, e.g., the language used, the words used, the topic, the home country of the newsgroup or TV station, etc. We specify a hierarchy of specific names for dialect regions (such as Maghreb>Tunisian and Levantine>Palestinian) and sub-dialectal features (such Urban or Bedouin). This allows a degree of reasonable approximation in case of doubt.

3. Annotation Experiments

In this section, we present some preliminary results on an annotation task using our guidelines. We first describe the annotation and detail statistics on the distribution of labels across different genres. Then we present inter-annotator agreement results on a portion of the annotated data.

3.1 Corpus Annotation

We annotated a small corpus of 59 documents (19,160 words) in four genres: newswire (NW), web text (WT), broadcast news (BN) and broadcast conversation (BC). The annotation was done as part of MT error analysis research (Kirchhoff et al. 2007), and our corpus choice was dictated by this task. NW and WT were naturally occurring text as opposed to BN and BC which have been transcribed by the Linguistic Data Consortium (LDC).[3] NW data came from Agence France-Presse, Xinhua News and Assabah. WT data came from different Google and Yahoo groups, such as IslamToday or YaMuslim. BC data came from different shows on AlJazeera and LBC. BN data came from AlJazeera, LBC and Dubai TV. Our annotator was a female Levantine Arabic speaker. The data overall was rather free of some of the dialectal phenomena that influenced many of our decisions (particularly dialectal orthographic inconsistencies): the transcripts of the BN and BC data were carefully produced according to LDC guidelines (Maamouri et al. 2004b); and the religious theme of the WT data made it more MSA-like than some other web texts we have seen which are much more dialectal. Thus, this set of texts is not truly representative of the types of dialect switching we expect to find.

Table 1 presents various statistics over the annotated set. The first four columns of data belong to the four genres annotated. The next two combine the two transcribed spoken genres (BN and BC) into BX, and the two written genres (WT and NW) into TX. The last column combines all the data. The first data row shows the number of documents. The second shows the number of segments. BX data had more segments (lines) that were shorter (average 8 words) than TX data (average 19). The next five rows show the distribution of segment level labels. Overall BX data has a larger number of segments in level 1 than level 0. This stands in stark contrast to the TX data whose segments are almost all in level 0. Within BX, BC exhibits a higher degree of level 1 than BN. This is a consistent trend with expectations about the genres: BC is less rehearsed and is more reactive (and thus more dialectal) as opposed to BN which is primarily read. Within TX data, we do not see the trend we expect; namely that WT data is more dialectal. This is perhaps a result of the data being of a higher MSA quality than is commonly the case in other news groups because the groups' themes are religious.

Next in Table 1 is a row showing the number of words per genre. This is followed by the distribution of the four word levels. The distributions here are consistent with what we expect: BC is more dialectal than BN than WT than NW. The jump in NW at level 3 is the result of the Levantine annotator considering the spelling of some month names as dialectal because they are different from her MSA: e.g., the name used for the month 'February' was فبراير *fibrAyir*, which is acceptable in Egypt as MSA but not in the Levant (corresponding MSA month name is شباط *šbAT*) (Omar 1976). We will revisit this issue when we discuss inter-annotator agreement in Section 3.2. Overall, at the word annotation, BX has less of level 0 and more of higher levels than TX.

	BC	BN	WT	NW	BX	TX	ALL
Doc's	6	8	17	28	14	45	59
Seg's	640	437	280	287	1077	567	1644
Level 0	27.7	57.4	98.9	96.2	39.7	97.5	59.7
Level 1	67.5	41.2	0.7	2.4	56.8	1.6	37.8
Level 2	3.1	0.5	0.4	1.4	2.0	0.9	1.6
Level 3	0.5	0.2	0.0	0.0	0.4	0.0	0.2
Level 4	1.3	0.7	0.0	0.0	1.0	0.0	0.7
Words	4619	3919	5042	5580	8538	10622	19160
Level 0	96.0	97.9	98.7	99.3	96.9	99.0	98.1
Level 1	2.2	1.5	1.2	0.2	1.9	0.7	1.2
Level 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Level 3	1.7	0.6	0.1	0.5	1.2	0.3	0.7

Table 1 Statistics of annotated corpus (levels areprecentages). BC is transcribed broadcast conversation, BN is transcribed broadcast news, WT is web texts (web discussion forums), and NW is newswire. BX is the combination of the two transcribed speech genres (BC and BN), while TX is a combination of the two text genres (WT and NW).

3.2 Inter-annotator Agreement

We double annotated around 20% of the data (11 documents) roughly equally distributed over all genres. Our second annotator is a female Egyptian Arabic speaker. We consider here word-level and segment-level annotations only. At the word level, the overall agreement (as accuracy) is over 98.5%. The most common label agreed on is *level 0* (97.1%). The corresponding Cohen's (1960) kappa score is 0.72 indicating a good degree of agreement. The largest disagreement is between levels 0 and 1. A large portion of the disagreement is the result of our guidelines' lack of specification of MSA standard spelling of some words, particularly proper names; although MSA is standard across the Arab world, some regional varieties of spellings exist. Our Egyptian and Levantine annotators disagreed on whether كوبنهاجن *kuwbinhAjin* 'Copenhagen' is level 0 or 1: it is level 0 in Egyptian MSA, but the preferred spelling in Levantine MSA is كوبنهاغن *kuwbinhAγin*. Similarly, the spelling of names of continents such as 'Africa' using a Taa Marbuta (افريقية *Afriyqyah̄*) was not accepted by the Egyptian annotator who preferred (افريقيا *AfriyqyA*). There is a small portion of levels 0 and 3 disagreement too. These were primarily the result of interpreting interjections literally (as MSA) or functionally (as dialect). For example, the use of والله *waAll~ah* '(lit. by God)' as a semantically empty interjection led one of our annotators to mark it as lexically dialectal (level 3). The segment-level annotation inter-annotator agreement is lower than word-level annotation: basic accuracy agreement is 78% and the kappa measure is 0.56.

[3] http://www.ldc.upenn.edu/

4. Conclusions and Future Work

We have presented a proposal for annotation guidelines identifying dialect switching between MSA and at least one dialect in written text. We have reported on some initial annotation experiments showing that dialect annotation has distinct distributions in different genres. Our initial results on inter-annotator agreement are encouraging. However, much more work is needed in clarifying and detailing the guidelines. In particular, the results of the inter-annotator agreement analysis suggest a need to address the existing variations of MSA in different regions in the guidelines to specify a reference point and/or make the annotators aware of these variations: we cannot have an annotator flag something as dialectal though it is perfectly acceptable MSA in another part of the Arabic-speaking world. Once the guidelines have been updated, we plan to annotate additional data of different variety in the future.

5. Acknowledgements

This work was partially funded by DARPA GALE contract HR0011-06-C-0023 and by NSF SGER grant BCS-0749062. We would like to thank the participants in the April 2008 workshop on the annotation of code switching for their useful feedback.

6. References

السامرائي, ابراهيم. (1981). العربية التونسية (فصل 13). التطور اللغوي التاريخي. دار الاندلس. بيروت.

نخله, رفائيل. (1959). غرائب اللهجه اللبنانية السورية. المطبعة الكاثوليكية. بيروت.

Badawi, S, and Hinds, M. (1986). *A Dictionary of Egyptian Arabic: Introduction*. Beirut: Librairie du Liban.

Bateson, Mary Catherine. (2003) Arabic Language Handbook. Georgetown University Press.

Bikel, Daniel. (2002). Design of a multi-lingual, parallel processing statistical parsing engine. In Proc. of International Conference on Human Language Technology.

Brustad, Kristen E. (2000). The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects. Georgetown University Press.

Buckwalter, Tim. (2004). Buckwalter Arabic morphological analyzer version 2.0. 12(3):373–418.

Butros, Albert. (1973). Turkish, Italian, and French Loanwords in the Colloquial Arabic of Palestine and Jordan. Studies in Linguistics. Volume 23.

Chiang, David, Mona Diab, Nizar Habash, Owen Rambow and Safi Sharif. (2006). Parsing Arabic Dialects. In Proc. of the European Chapter of the Association for Computational Linguistics (ACL). Trento, Italy.

Cohen, Jacob. 1960. *A coefficient of agreement for nominal scales*. Educational and Psychological Measurement, 20(1):37-46.

Diab Mona and Nizar Habash. (2007). Arabic dialect processing tutorial. The conference of the North American Chapter of ACL, Rochester, NY.

Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. (2007) “Automated Methods for Processing Arabic Text: From Tokenization to Base Phrase Chunking”. Book Chapter. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Editors Antal van den Bosch and Abdelhadi Soudi. Kluwer/Springer Publications.

Ferguson, Charles A. (1959). "Diglossia". *Word 15,* pp. 325--340.

Habash, Nizar, Abdelhadi Soudi and Tim Buckwalter. (2007). “On Arabic Transliteration.” Book Chapter. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Editors A. van den Bosch and A. Soudi. Kluwer/Springer Publications.

Habash, Nizar and Owen Rambow. (2005). Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop. In Proc. of ACL. Ann Arbor, MI.

Habash Nizar and Owen Rambow: (2006). MAGEAD: a morphological analyzer and generator for the Arabic dialects. In Proc. of ACL. Sydney.

Habash, Nizar. (2006). “On Arabic and its Dialects,” Multilingual Magazine. #81 Volume 17 Issue 5.

Holes, Clive. (2004). Modern Arabic: Structures, Functions, and Varieties. Georgetown University Press.

Kirchhoff, Katrin, Owen Rambow, Nizar Habash, and Mona Diab. (2007). Semi-automatic error analysis for large-scale statistical machine translation. In Proc. of MT Summit XI, Copenhagen, Denmark.

Maamouri, Mohamed, Ann Bies, and Tim Buckwalter. (2004b). The Penn Arabic Treebank: Building a largescale annotated Arabic corpus. In NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt.

Maamouri, Mohamed, Tim Buckwalter, and Christopher Cieri. (2004b). Dialectal Arabic Telephone Speech Corpus: Principles, Tool Design, and Transcription Conventions. In NEMLAR.

Maamouri, Mohamed, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. (2006). Developing and using a pilot dialectal Arabic tree-bank. In Proc. of LREC’06.

Omar, Margaret. (1976). Levantine and Egyptian Arabic: Comparative Study. Foreign Service Institute. Basic Course Series.

Information Retrieval in Arabic Language

Malek Boualem (1)   Ramzi Abbes (2)

(1) France Telecom Orange Labs, France
Email : malek.boualem@orange-ftgroup.com

(2) Lyon 2 University / ICAR-CNRS, France
Email : ramzi.abbes@univ-lyon2.fr

Abstract

Web search engines provide quite good results for Latin characters-based languages. However, they still show many weaknesses when searching in other languages such as Arabic. This paper discusses a qualitative analysis of information retrieval in Arabic, highlighting some of the numerous limitations of available search engines, mainly when they are not properly adapted to the Arabic language features. To support our analysis we present some results based on thorough observations about various Arabic linguistic phenomena. To validate these observations, we mainly have tested the Google search engine. Arabic information retrieval still faces many difficulties due to the Arabic linguistic features, especially its complex morphology and the absence of vowels in available documents and texts. These specificities often cause significant dissymmetry between the indexation process and the query analysis. We present in this paper some of the morphological constraints of Arabic language and we show through experimental tests how search engines deal with them. Finally this paper clearly states that information retrieval in Arabic language will never succeed without including language processing tools at all the linguistic levels (lexical, syntactic and semantic).

Keywords: Information retrieval, Arabic language, Google

1 Introduction

With 90,83% of the internet users (see Figure 1, December 2007[1]), Google is probably the most powerful search engine on the market, or more precisely the most used one, because there is correlation between these two aspects. Indeed the Google PageRank algorithms are very sensitive to the user's behaviour (Brin; Page, 1988). These algorithms balance positively or negatively web pages according to the click numbers on the corresponding links and the PageRank scores web pages according to the number of hypertext links they contain (Chen & al., 2007). This observation is also very accurate when using Google to search in Arabic language. For example, we have noticed that most of the top list answers correspond to various forums منتديات or to some other specific information sources. The Google PageRank attributes higher scores to these information sources as they contain a high number of hypertext links and also because they are more commonly used in the Arabic world instead of other information sources such as scientific papers or publications.

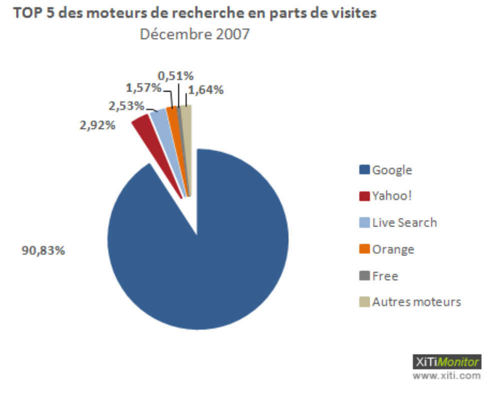


Figure 1. Top 5 of search engines based on the user's number

Most of the Arabic internet users master a second language, mainly English or French. As the information on the Web is widely available in these languages, the Arabic internet users often prefer searching in these languages rather than in Arabic. Of course consequently this situation does not help the development of Arabic information resources on the Web.

Moreover to avoid editing problems of Arabic texts on various screens and operating systems, a lot of publishers (e.g. newspapers) provide the Web with PDF documents. This situation also does not help searching information in Arabic language.

[1] http://barometre.secrets2moteurs.com/index.php/Barometre-1ere-position-xiti

2 Arabic language on the Web

2.1 Dissymmetry between indexation and query analysis processes

Searching in Arabic language meets a fundamental problem related to a certain dissymmetry between the indexation side and the query analysis side. One of the reasons is related to the Arabic vowels. For example, in the indexation process, the verb "to write" (كَتَب) together with the noun "books" (كُتُب) are indexed under the same entry كتب, since they probably do not have vowels in the Arabic original text. This problem concerns most of the Arabic verbs and nouns that are based on three-letter roots, like the word "شعر" which can have different meanings depending on the different combinations of vowels ("to feel", "poetry", "hair", etc.) or the word "علم" which can mean "flag", "science", etc.

Another reason of this dissymmetry is related to the agglutination feature of Arabic words. The agglutination happens when a minimal form of a word is attached to various proclitics (interrogative style, similarity, link, etc.) or to various enclitics (mainly to add pronouns). These three examples can illustrate various situations of agglutination: (1) Kateb Yasseen كاتب يسين, (2) Does Kateb Yasseen ? أكاتب يسين, (3) I write to Yasseen أكاتب يسين.

2.2 Information retrieval in Arabic language

Most of the search queries, whatever the languages are, concern named entities such as proper names, etc. In another hand, to check the linguistic structures of the queries, we have made some tests using a sub-set containing 2850 Arabic queries that have been submitted to a multilingual Web directory described in (Boualem & al., 2001). These tests allowed us to see that 94,2% of the queries concern nominal structures, only 3,30% concern verbal structures and only 2,5% concern grammatical words. In fact these results can be lightly adjusted if we consider that queries do not contain vowels. Thus, apart some verbal structures and some non-ambiguous grammatical words such as من, متى, اخترع , most of the queries are very ambiguous (نزل, رقص, طلب , etc).

Concerning Arabic proper nouns, they often are derived from verbal structures (active participle, passive participle, etc.). For example كاتب means "author" and also is a part of a proper name such as in Kateb Yasseen. However, searching كاتب generally retrieves "author".

To argue these observations we present here some examples of search queries using Google :

- for the keyword كَتَب (with vowels such as "katab"), the first results are related to "books" (which transcription is "kutub") : in this case can we consider these results as a consequence of the "ranking" algorithm or is it related to a kind of "priority" for nouns ? Anyhow we can easily see that adding vowels to the keywords has no influence on the searching results.

- when searching for **جمال الدين الأفغاني** or **جمال عبد الناصر الزعيم**, the keyword جمال first retrieves adjectival results related to "beauty", and not related to the proper noun جمال. More precisely, we get 5 340 000 answers for **جمال**, 737 000 answers for **جمال الدين**, and 70 700 answers for **جمال الدين الأفغاني**. When searching for **ناصر** we get 805 000 answers for **جمال عبد**, 293 000 answers for **جمال عبد الناصر** and 253 000 answers for **جمال عبد الناصر الزعيم**. In the same way, the keyword الزعيم, retrievers 2 100 000 answers for **الزعيم**. In this case we noticed that the first displayed results are related to some soccer blogs, or related to theatre information about Adel Imam (742 000 answers for **الزعيم عادل**). Hence the first result related to **جمال عبد الناصر** comes in the 30th position. We conclude that there is a significant lack in processing named entities.

3 Benefits of natural language processing for information retrieval in Arabic language

3.1 Lemmatisation

We think that information retrieval is somehow language-dependent in the sense that search engines should adapt indexation and searching strategies to the language specificities. For Arabic, which has a complex (even regular) morphology (Dichy, 1990), we think that search engines should primary focus at least on lemmatisation. We try here, through some examples of some linguistic phenomena, to show the limitations of "artificial linguistic processing" in the indexation process and the benefits of lemmatisation for information retrieval in Arabic language.

Arabic has a very flexional morphology where morphological families can reach huge numbers of combinations. A lot of graphical forms of words, even they seem very similar, might not belong to the same semantic families and even to the same morphological families. Let us see for example the search results for the derived forms of the word قال : the query « *قال* » provides more than 146 forms, which largely exceeds the derivational combinations of this word. Indeed, the query retrieves words such us الانتقال , اعتقالهم , استقالة , العقال , مقاليد , اقالتهم , التقاليد , برتقالية , اثقالاً , وقالباً , الأقاليم , قالب , … Moreover besides this huge "noise", a lot of other morphological variations of this word need to be found through other queries (e.g. imperfective يقول and other related deverbal forms).

For another example with the keyword « سماء », Google retrieves 594 000 answers by applying a completion method. Results also contain 279 000 answers for **أسماء**, where we can also find **الأسماء** (which is a rare plural form of سماء used for example in titles such as أسماء السماوات.). We think that these morphological and semantic distances are due to the fact of applying to Arabic language the lemmatisation rules of other languages such as English.

In another hand, even applying Arabic lemmatisation rules does not allow obtaining good results in information retrieval because the derivational system of Arabic is more complex than just using suffixes. However lemmatisation in searching Arabic is necessary due to the agglutinant specificities of Arabic words completed by using proclitics and enclitics.

3.2 Gender, number and lemmatisation

To enrich our analysis about lemmatisation in searching Arabic we focus now on two nominal aspects, gender and number and we try to show the limitations of indexing techniques.

3.2.1 Singular and plural

Let us consider the plural word كتابات, the "standard" lemmatisation procedure, which consider making the plural form by adding the ات suffix to the singular form, should give the lemma كتاب , but the right lemma is كتابة. The same problem can be found when trying to lemmatise the dual form فتاتان to the singular form فتات, but the right one is فتاة. Our tests on Google have shown its limitations in processing theses kind of linguistic phenomena when confusing word terminations ت and ة. The "broken plural", which is a non-regular plural in Arabic language and that does not follow any flexional rules, comes to add more complexity to the lemmatisation procedures (رجال-رجل, for man-men and نساء-نسوة-امرأة for woman-women). Also some dual forms, in a morphological point of view, might correspond to singular forms, such as for example the country noun البحرين or the personal pronoun محمدين.

We also have analyzed the user's queries and have extracted the following information about using singular, dual and plural forms in keywords :

Number				
Singular	Dual	Plural		
74,21%	1,77%	24,02%		
		Regular masculine	Regular feminine	Broken plural
		71,09%	21,29%	7,52%

3.2.2 Masculine and feminine

Suffixation rules in general can be used to obtain masculine and feminine forms. To obtain a feminine form, the "standard" rule aims to add the suffix ة to the masculine form. However (again) this rule can not be always systemised, such as in the feminine words اثارة and دراسة which do not have masculine forms. Also there are many masculine Arabic word ended by the letter ة, such as in the word خليفة. In another hand, the gender might also be expressed through different words having different roots, like for example these masculine forms رجل أب ولد حصان جمل.

We also have analyzed the user's queries and have extracted the following information about using masculine and feminine in keywords :

Gender					
Masculine	Feminine				
50,13%	49,84%				
	With suffix ة		Without suffix ة		Others
	with masculine	without masculine	with masculine	without masculine	Feminine of masculine plural
	47,11%	11,69%	16,81%	1,01%	23,38%

4 Conclusion

Arabic information retrieval still faces many difficulties due to the Arabic linguistic features, especially its complex morphology and the absence of vowels in available documents and texts. These specificities often cause significant dissymmetry between the indexation process and the query analysis. We have presented in this paper some of the morphological constraints of Arabic language and we have shown through experimental tests how search engines deal with them. Finally this paper clearly states that information retrieval in Arabic language will never succeed without including language processing tools at all the linguistic levels (lexical, syntactic and semantic).

5 References

Abbes, R. (2004). La conception et la réalisation d’un concordancier pour l’arabe. Thèse de doctorat en Sciences de l’Information, Lyon : INSA, décembre 2004.

Abbes, R. & Dichy, J. (2008). Extraction automatique de fréquences lexicales en arabe et analyse d’un corpus journalistique avec le logiciel AraConc et la base de connaissances DIINAR.1, JADT’2008, 12-13 mars 2008, ENS-LSH Lyon, France.

Abbes, R. & al. (2004). The Architecture of a Standard Arabic Lexical database: some figures, ratios and categories from the DIINAR.1 source program. In : COLING'04, 20th International Conference on Computational Linguistics. Proceedings of the Workshop Computational Approaches to Arabic Script-bases Languages, 28.08.2004, Genève : 15-22.

Buckwalter, T. (2004). Issues in Arabic Orthography and Morphological Analysis. In : COLING'04, 20th International Conference on Computational Linguistics. Proceedings of the Workshop Computational Approaches to Arabic Script-bases Languages, 28.08.2004, Genève : 31-41.

Boualem, M. & Sneifer, R. (2001). Hahooa Arabic Web Directory & Natural Language Processing for Arabic Information Retrieval, ACL 2001, Workshop on Arabic Language Processing, Toulouse, France, July 2001.

Boualem, M. (1995). Arabic language processing, SNLP'95 Symposium on Natural Language Processing, Bangkok, Thailande, Août 1995.

Boualem, M. & Zajac, R. (1999). Unicode-based Arabic text, ATLAS'99, Arabic Translation and Localisation Symposium, Tunis, May 26-28, 1999.

Brin, S. & Page, L. (1988). The anatomy of a large-scale hypertextual web search engine, Computer Networks and ISDN Systems 30 (1988), pp. 107–117.

Chen, P. & al. (2007). Finding scientific gems with Google's PageRank algorithm, Journal of InformetricsVolume 1, Issue 1, January 2007, pp. 8-15.

Dichy, J. (1990). L'écriture dans la représentation de la langue : la lettre et le mot en arabe. Thèse d'État, Université Lumière-Lyon 2.

Harman, D. (1991). How effective is suffixing? Journal of the American Society of Information Science. vol. 42, No 1. pp. 7-15, 1991.

Memory-Based Vocalization of Arabic

Sandra Kübler, Emad Mohamed

Indiana University
Department of Linguistics
1021 E 3rd St.
Bloomington, IN-47405
USA
{skuebler,emohamed}@indiana.edu

Abstract

The problem of vocalization, or diacritization, is essential to many tasks in Arabic NLP. Arabic is generally written without the short vowels, which leads to one written form having several pronunciations with each pronunciation carrying its own meaning(s). In the experiments reported here, we define vocalization as a classification problem in which we decide for each character in the unvocalized word whether it is followed by a short vowel. We investigate the importance of different types of context. Our results show that the combination of using memory-based learning with only a word internal context leads to a word error rate of 6.64% on newswire text. However, if punctuation and numbers are excluded from vocalization, the best results are reached by including the left context. On the data set by Zitouni et al. (2006), our best performing system reached a word error rate of 17.5%.

1. Introduction

The problem of vocalization, or diacritization, is essential to many tasks in Arabic NLP. Arabic is generally written without the short vowels, which leads to one written form having several pronunciations with each pronunciation carrying its own meaning(s). The word form 'mskn' is an example for a highly ambiguous word. Its possible pronunciations include 'maskan' (home), 'musakkin' (analgesic), 'masakn' (they-*fem*. have held), or 'musikn' (they-*fem*. have been held). The importance of vocalization become clear when we look at how Google Translate renders 'A$tryt Almskn mn AlSydlyp' (I bought a pain killer from the pharmacy): as 'I bought the home from the pharmacy'. This error would not occur if the input to the translation system were vocalized in a first step before the actual translation process. However, vocalization is far from trivial: the example above shows that the vocalized words of a single unvocalized form differ in their parts-of-speech as well as in their meaning. This shows that vocalization performs implicit POS tagging and word sense disambiguation. It is also obvious that word forms cannot be vocalized in isolation, the task is heavily dependent on the context of the word.

In the experiments reported here, we investigate the importance of different types of context for vocalization. We follow Zitouni et al. (2006) in defining vocalization as a classification problem in which we decide for each character in the unvocalized word whether it is followed by a short vowel. Additionally, the classification task includes the shadda and sokoon (lack of a vowel). The shadda is consonant gemination and is usually found in combination with another vowel, thus resulting in 3 classes, $\tilde{a}, \tilde{\imath}, \tilde{u}$. The shadda plays an important role in the interpretation of Arabic words because it is used, inter alia, to discriminate between the base form of a verb and its causative form: *kataba* (to write), *kat˜aba* (to make write). At present, we ignore case endings, mood endings, and nunation (syntactic indefiniteness marker for a noun or adjective).

We investigate here how well the task can be performed if only context from the same word is available as compared to having access to a lexical context of 5 words on each side. We also investigate which types of features are the most important ones. Lastly, we investigate the learning curve to determine how much training data we need for reliable results.

2. Previous Research

The first approaches to the vocalization of Arabic defined the problem word-based, i.e. the task was to determine for each word the complete vocalized form. Gal (2002) uses a bigram HMM model for vocalizing the Qur'an and achieves a word error rate (WER) of 14%. His error analysis showed that the errors resulted mostly from unknown words. Kirchhoff et al. (2002) use vocalization in speech recognition. Their system uses a unigram model extended by a heuristic for unknown words, which retrieves the most similar unlexicalized word and then applies edit distance operations to turn it into the unknown word. They reach a WER (for vocalization) of 16.5% on conversational Arabic. Nelken and Shieber (2005) tackle the problem with weighted finite-state transducers. For known words, morphological units are used for retrieving the vocalization while unknown words are vocalized based on the sequence of characters. They reach a WER of 12.8%. Zitouni et al. (2006) use a maximum entropy model in combination with a character based classification. Their features are based on single characters of the focus word, morphological segments, and POS tags. They reach a WER of 17.9%. Habash and Rambow(2007) perform a full diacritization including case endings and nunation. They use the Buckwalter analyzer (Buckwalter, 2004) to obtain all possible morphological analyses, including all diacritics. Then they train individual classifiers to disambiguate between these analyses. Residual ambiguity is resolved via an n-gram language model. Habash and Rambow reach a WER of 14.9% on the test set of Zitouni et al. (2006).

A comparison of the different approaches shows that the definition of vocalization as inserting vowels between char-

w_{-5}	w_{-4}	w_{-3}	w_{-2}	w_{-1}	c_{-5}	c_{-4}	c_{-3}	c_{-2}	c_{-1}	c	c_1	c_2	c_3	c_4	c_5	w_1	w_2	w_3	w_4	w_5	v
kl	$y”	tgyr	fy	HyAp	_	_	_	_	_	A	l	m	t	$	r	styfn	knt	EndmA	Evrt	Ely	_
kl	$y”	tgyr	fy	HyAp	_	_	_	_	A	l	m	t	$	r	d	styfn	knt	EndmA	Evrt	Ely	_
kl	$y”	tgyr	fy	HyAp	_	_	_	A	l	m	t	$	r	d	_	styfn	knt	EndmA	Evrt	Ely	u
kl	$y”	tgyr	fy	HyAp	_	_	A	l	m	t	$	r	d	_	_	styfn	knt	EndmA	Evrt	Ely	a
kl	$y”	tgyr	fy	HyAp	_	A	l	m	t	$	r	d	_	_	_	styfn	knt	EndmA	Evrt	Ely	a
kl	$y”	tgyr	fy	HyAp	A	l	m	t	$	r	d	_	_	_	_	styfn	knt	EndmA	Evrt	Ely	ĩ

Table 1: The word ’Almt$rd’ represented with one instance per word; the class represents the vowel to be inserted after the character.

acters results in the lowest WER. However, these studies leave the lexical context of words for the most part unexplored. In the present study, we will investigate this area of research.

3. Experimental Setup

3.1. Data

We use the Penn Arabic Treebank (Bies and Maamouri, 2003) as the data source. The treebank is encoded in Buckwalter transliteration (Buckwalter, 2004) and is available in a vocalized and an unvocalized version. From the treebank, we extracted 170 000 words from the AFP section (part 1 version 2.0) and approximately 160 000 words from the Ummah section (part 2 version 2.0). In order to assure that our results can be compared with the results of Zitouni et al., we performed additional experiments on their data set, which is taken from part 3, version 1.0. This data set contains news items from the An Nahar News text, it is split into a training set of approximately 288 000 words and a devtest set of approximately 52 000 words.

As mentioned previously, we define vocalization as a classification problem: For each character in the focus word, the learner needs to decide whether the character is followed by a short vowel and what the short vowel is. We will call this character the focus charcter. The task also involves the restoration of the shadda (gemination).

The features used for determining the short vowel following the focus character consist of the focus character itself (c), its local context in terms of neighboring characters within the focus word, and a more global context of neighboring words. For the local context, 5 characters to the left ($c_{-5} \ldots c_{-1}$), 5 characters to the right ($c_1 \ldots c_5$) are used; for the lexical context, 5 words to the left ($w_{-5} \ldots w_{-1}$), and 5 words to the right ($w_1 \ldots w_5$). The last value in the vector (v) provides the correct classification, i.e. the short vowel to be inserted after c, or - in cases where no vowel is inserted in that position. The instance for the Arabic word ’Almt$rd’, for example, is shown in Table 1. The last instance contains an example of the shadda, represented by the tilde sign.

For the experiments to determine the effect of context on vocalization accuracy, we used 10-fold cross validation. For the experiments concerning the size of the training set and the ones with the data set from Zitouni et al., we used a designated test set. For the 10-fold CV experiments, we did not build the folds randomly but rather sequentially, thus ensuring that a single fold contains consecutive articles, which may cover different topics from the other folds. However, we made sure that all instances of a word were put in the same fold. For the learning curve experiments, we selected one fold that showed an average accuracy as test set.

3.2. Methods

For classification, we used a memory-based learner, TiMBL (Daelemans et al., 2007). Memory-based learning is a lazy-learning paradigm, which assumes that learning does not consist of abstraction of the training instances into rules or probabilities. Instead, the learner uses the training instances directly. As a consequence, training consists in storing the instances in an instance base, and classification finds the k nearest neighbor in the instance base and chooses their most frequent class as the class for the new instance. Memory-based learning has been proven to have a suitable bias for many NLP problems (Daelemans et al., 1999). One of the reasons for this success is that natural language exhibits a high percentage of subregularities or irregularities, which cannot be distinguished from noise. Eager learning paradigms smooth over all these cases while memory-based learning still has access to the original instance. Thus, if a new instance is similar enough to one of these irregular instances, it can be correctly classified as such.

Memory-based learning was chosen for two reasons: First, this approach weights features based on information gain or gain ratio (Daelemans et al., 2007), thus giving some indication of the most and the least important features. Additionally, it is a paradigm that is capable of handling symbolic features with a high number of different feature values. This allows us to use complete context words as features.

Parameter settings for TiMBL were determined first. The best results are obtained for all experiments with the IB1 algorithm with similarity computed as weighted overlap, i.e. with a standard city block metric as distance measure. Relevance weights are computed with gain ratio, and the number of k nearest neighbors (or in TiMBL’s case, nearest distances) is set to 1. The latter setting is noteworthy in that it signals that only the closest training examples provide reliable information for classifying a character. Normally, higher values of k are beneficial, they provide a certain smoothing factor.

For evaluation, we calculate the error rate based on characters (CER) and based on words (WER). A decision of the classifier is considered correct if both the vowel to be inserted and the shadda (if present) are correct. Since previous work has been evaluated on all words in the texts,

	with punctuation		w/o punctuation	
	CER	WER	CER	WER
baseline	–	47.20	–	54.45
character context	2.22	6.64	6.20	14.40
left word context	2.26	7.06	2.33	7.57
word context	2.35	6.86	2.64	9.91

Table 2: The results of the vocalization experiments with TiMBL.

including punctuation and other non-vocalized words such as numbers, we will present these results to allow a comparison to previous work. However, we believe that words that are never vocalized should not be considered in the evaluation. For this reason, we also provide results where such words were present in the classification, but were excluded in the evaluation.

4. Results

The results of our experiments with regard to different contexts as well the baseline are shown in Table 2. The baseline experiment was set up so that the classifier was presented with 11 words: the focus word, 5 context words to its left, and 5 context words to its right. The results for the baseline show that vocalization is a difficult task, even in our data set where a word on average has only 1.67 vocalizations. This figure is considerably lower than the average on normal texts. Debili et al. (2002) found that on average, each unvocalized word type has 2.9 vocalized versions, and there is an average of 11.6 vocalized versions per word token in a text. We assume that the difference is a consequence of the different text genres.

4.1. Relevant features

While the baseline shown in Table 2 presents results for vocalizing the whole word, the following three lines report the results for the experiments in which we define the task as deciding for each character whether it is followed by a vowel. The left half of the table report the standard evaluation, which includes punctuation and numbers as words. In the right half, we performed the evaluation only on words that can be vocalized. This leads to higher error rates since the classification of punctuation is trivial.

The experiment in line 2 uses only a character context of 5 characters to each side of the focus character but ignores the context words, i.e. the features from c_{-5} to c_5 in Figure 1 are used. The next experiment uses the lexical context to the left of the focus word in addition to the character context but ignores the context words on the right, i.e. the features from w_{-5} to c_5 are used. Finally, the last experiment uses all features shown in Figure 1, i.e. it uses the character context as well as the lexical features to the left and to the right of the focus word. When going from classifying complete words to classifying characters separately, the results improve dramatically. This method results in a WER of 6.64% in standard evaluation (including punctuation).

Surprisingly, adding the context words does not improve the classification results when punctuation is included in the evaluation. On the contrary, it results in a lower WER. This is unexpected, we would have expected that at least in cases where the vocalizations have different parts of speech, the lexical context would provide important information. One possible explanation for these negative results may be data sparseness. However, if we use the lexical context on both sides of the focus word, the CER is lower but the WER is higher than in the experiment with the left context only. This shows that the individual decisions concerning single vowels become more difficult but the recognition of complete words becomes more stable. Thus, in some cases, the lexical context does improve classification. This also becomes evident when we compare the results of single folds in the 10-fold setting. Some of the folds have better results in the left context setting, and some in the full context setting.

When we exclude punctuation and other words that cannot be vocalized from the evaluation, it becomes obvious that the task is much more complex. The WER deteriorates from 6.64% (evaluated on all words) to 14.4% (only words that can be vocalized) in the experiment when only word internal features are used. However, when we compare the difference in WER between the experiment with only word internal features and the experiments with context features, the results are better for the experiment with the complete context information and even better for the left context only. Therefore, we have to conclude that the context does provide valuable information, but this is obscured by including words without vowels.

Next, we look at the weights that TiMBL assigns to the different features in the character based experiments. Here, the results are very stable. If we look at the gain ratio weights, in all experiments over all folds, we find the same ordering of features. The feature with the highest weight is the character following the focus character, c_1. The next most important feature is the focus character, c. The third most important character is the next character to the left, c_{-1}, followed by all its preceding characters c_{-5} to c_{-2}, followed by all the characters to the right of c_1: c_2, c_5, c_3, and c_4. Why c_5 is more important than its preceding characters remains unclear. One would expect that syllable structure might be an influence here. But since we use a sliding window approach, the features do not correspond to fixed syllabic constituents.

4.2. Comparison to Previous Work

In order to present results that are comparable to the results by Zitouni et al. (2006) and Habash and Rambow (2007), we trained and tested our classifier with the best parameter and feature settings on the data sets as defined by Zitouni et al. This means, for training, we used the first 85% of the files of the Penn Arabic Treebank part 3, version 1.0, and the last 15% for testing. The best settings for TiMBL were selected based on the evaluation with punctuation since Zitouni et al. and Habash and Rambow used this evaluation. The results of this experiment are shown in Table 3. We also conducted the evaluation without punctuation. In this case, the classifier reached a CER of 5.7% and a WER of 20.49%.

The results of this evaluation show that this part of the Penn Arabic treebank is considerable more difficult to vocalize than our initial set. The first attempt at explaining this dif-

	CER	WER
Zitouni et al. (lex. features)	8.2	25.1
Zitouni et al. (lex. + segment features)	5.8	18.8
Zitouni et al. (all features)	5.5	18.0
MBL	5.7	17.5
Habash and Rambow	4.8	14.9

Table 3: The results of the vocalization experiments on the data set of Zitouni et al.

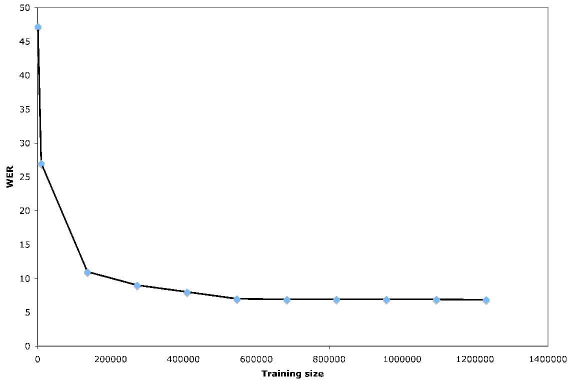


Figure 1: The learning curve for word internal context.

ference would be the assumption that the number of vocalized forms per unvocalized word is higher. This is not the case, the number is 1.22 and thus lower than for the data set used in the previous section. However, if we look at the unknown words (types), we find a huge difference in the data sets: In the newswire texts used in the previous section, the percentage of unknown words in the test set is 28.86%. In the test set by Zitouni et al., the percentage is 41.12%.

Surprisingly, our results are comparable to the ones by Zitouni et al. when using all their features. Since our best feature setting as determined above contains only word internal information without any further linguistic processing, the memory-based learner only has access to surface forms. This leads to our preliminary conclusion that memory-based learning is capable of extracting more information from the pure surface forms than other learners. However, our results are considerable lower than the best published results on this data set (Habash and Rambow, 2007), which reached a WER of 14.9%. However, their system performs a complete morphological analysis and thus uses more linguistic knowledge than ours.

4.3. Size of the training set

The next question to investigate concerns the importance of the training set size. In order to investigate how much training data we need for the task, we conducted an experiment in which we started with a small training set containing 1 000 character instances, and then continually increased the training set size to the full training set size of 1 230 723 character instances from the experiment in Section 4.1. The test set was kept stable, we used one of the folds for testing. In order to ensure reliable results, we chose a fold that resulted in average results in the ten-fold experiments reported in Section 4.1. All the experiments were performed with the best feature set determined in the experiments reported in this section, i.e. with the newscast data set and with characters from the focus word as the only features. The learning curve is shown in Figure 1. When training on a set of only 1 000 characters, the WER is 47.2%, but raising the size of the training set to 10 000 instances reduces the WER to 27.1%. The saturation point is reached at approximately 700 000 characters (which corresponds to 5 folds), with a WER of 6.9%. After this point, there are only minor improvements, and the WER reaches 6.64% for the whole training set.

5. Conclusion and Future Work

In the experiments reported here, we have investigated the vocalization of Arabic. The results show that the word internal context provides enough information for vocalizing a high percentage of words correctly. The best parameter and feature setting results in an error rate of 6.64% on newswire texts. Adding lexical context as additional features did not increase the performance of the memory-based classifier TiMBL when the evaluation is performed on all words. However, if punctuation and numbers are excluded from vocalization, the best results of 7.57% WER are reached by including the left context. Interestingly, the most informative feature is the character following the focus character although in general, the left character context within the focus word is more informative than the right character context. The learning curve shows that at least in the experiments with features only from within the focus word, a training set of 700 000 characters is sufficient for reliable results. For the future, we are planning to use a stemmer for Arabic to reduce the lexical features to stems in order to alleviate the sparse data problem concerning the lexical features. Additionally, we will follow Zitouni et al. (2006) and include part of speech information for all the words as well. Since the tagset of the Penn Arabic treebank is rather fine grained, we expect to reach the best results by reducing the tagset to a manageable level, following Diab (2007). Further lines of investigation concern the use of previous classification within a word for the classification of the next character and the use of automatically vocalized text for POS tagging.

6. References

Ann Bies and Mohamed Maamouri. 2003. Penn Arabic Treebank guidelines. Technical report, LDC, University of Pennsylvania.

Tim Buckwalter. 2004. Arabic morphological analyzer version 2.0. Linguistic Data Consortium.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–43. Special Issue on Natural Language Learning.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2007. TiMBL: Tilburg memory based learner – version 6.1 – reference guide. Technical Report ILK 07-07, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University.

Fahti Debili, Hadhebi Achour, and Emna Souissi. 2002. De l'etiquetage grammatical a la voyellation automatique de l'arabe. Technical report, Correspondances de l'Institut de Recherche sur le Maghreb Contemporain.

Mona Diab. 2007. Towards an optimal POS tag set for Arabic processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2007*, pages 157–161, Borovets, Bulgaria.

Ya'akov Gal. 2002. An HMM approach to vowel restoration in Arabic and Hebrew. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA.

Nizar Habash and Owen Rambow. 2007. Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007)*, Rochester, NY.

Katrin Kirchhoff, Jeff Bilmes, John Henderson, Richard Schwartz, Mohamed Noamany, Pat Schone, Gang Ji, Sourin Das, Melissa Egan, Feng He, Dimitra Vergyri, Daben Liu, and Nicolae Duta. 2002. Novel speech recognition models for Arabic - final report of the JHU summer workshop. Technical report, Johns Hopkins University.

Rani Nelken and Stuart Shieber. 2005. Arabic diacritization using weighted finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, MI.

Imed Zitouni, Jeffrey Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of Arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, COLING-ACL-2006*, Sydney, Australia.

Towards a Human-Machine Spoken Dialogue in Arabic

Younès BAHOU, Lamia HADRICH BELGUITH and Abdelmajid BEN HAMADOU

LARIS - MIRACL Laboratory
Faculty of Economic Sciences and Management of Sfax
B.P. 1088, 3018 - Sfax – TUNISIA
Phone (216) 74 278 777, Fax (216) 74 279 139
bahou.younes@caramail.com ; l.belguith@fsegs.rnu.tn ; abdelmajid.benhamadou@isimsf.rnu.tn

Abstract

This work is a part of automatic spontaneous Arabic speech understanding. We propose in this paper an analysis method guided by semantics and based on the frame grammar formalism. This method allows to represent meaningful oral utterances in semantic frame forms. It has the advantage of being robust when faced to analysis problems due to the spontaneity of interaction and the speech recognition limits. In this paper, we present our system of automatic Arabic speech understanding applied to the domain of Tunisian railway information. The understanding module of this system is based on the proposed method.

1. Introduction

Currently, there are several operational spoken dialogue systems. We cite as examples the Jupiter system for weather information in English (Zue et al., 2000) and the Ritel system for general information in French (Van Schooten et al., 2007). To our knowledge, there are no Arabic spoken dialogue systems which could communicate either in modern standard Arabic or in any other Arabic dialect. This is mainly due to the lack of tools that are necessary for the development of such systems. Indeed, most available tools represent small prototypes developed in research laboratories. We cite as examples the Satori (2007) tool for the recognition of Arabic numerals based on CMUSphinx (Satori et al., 2007a ; Satori et al., 2007b) and Ramsay (2008) tool for Arabic speech synthesis (Ramsay & Mansour, 2008).

In addition, few researchers have been interested in Arabic speech understanding which plays an important role in spoken dialogue systems. It involves extracting the meaning of the oral utterances which are inherently uncertain and ambiguous.

In this paper we propose our method of spontaneous Arabic speech understanding based on the frame grammar formalism (Bruce, 1975), which is inspired from case grammars (Fillmore, 1968 ; Fillmore, 1977), for the semantic representations of oral utterances. As an application domain, we chose the case of an interactive Arabic vocal server dedicated to information about Tunisian national railway company. This vocal server allows the user to interact with the machine using the Arabic speech to obtain railway information (e.g. train schedule fares, ticket reservation, etc.).

This paper focuses on five main parts. Section 2 details the difficulties of spontaneous Arabic speech understanding. Section 3 presents a brief overview of speech understanding approaches. Section 4 proposes our method of Arabic oral utterances understanding. Section 5 describes our SARF system (Arabic vocal server of railway information). Section 6 presents the implementation and evaluation of SARF understanding module.

2. Difficulties of Spontaneous Arabic Speech Understanding

The main goal of a speech understanding system is the construction of the semantic representations of the utterance given by the user. This representation is calculated by the sequence of recognized words, the dialogue history and the context (dialogue situations). A reliable and robust analysis has to face numerous problems that are due to the uncertain nature of oral utterances, the errors produced by the speech recognition module, the spontaneous nature of the interaction, etc.

In this section, we first describe the main difficulties of the understanding related to the characteristics of spontaneous spoken language (Minker & Bennacef, 2005). Then we present difficulties related to the spoken Arabic language.

The intrinsic characteristics of spoken dialogue that affect the automatic understanding (regardless the language) are mainly:

- The spontaneity of utterances that may contain redundant information, repetitions, repairs (self-correction), hesitations, false starts, etc.
- The ungrammatical structural utterance related not only to the spontaneity of the interaction but also to the spoken language itself.

In addition to these characteristics, spoken Arabic language presents a hierarchy of several varieties. First, we distinguish the modern standard Arabic which is the language of the holy Koran and it is used as the written language in all Arabic-speaking countries. Indeed, modern standard Arabic is used as the official language in all Arab countries. It is also the language used in official communications, and in most administrative and scientific documents. Second, there are the Arabic dialects that vary from one country to another and even from one region to another within the same country. Issued from modern standard Arabic, the respective grammatical systems of Arabic dialects differentiate with those of modern standard Arabic. Differentiation mainly concerns prosody, phonology, morphology and syntax. But despite this diversity, Arab societies feel that they belong to a homogeneous linguistic community. They are clung to the integrity of their language. Thus, we can see the importance of modern standard Arabic language, which is the common language for this population.

Note that in the context of this work, we are interested in modern standard Arabic for three reasons: **i)** it is understandable and used in all Arab countries **ii)** it is difficult for a semantic analyzer to handle different dialects **iii)** the absence of tools for Arabic dialects.

In addition to the varieties of spoken Arabic language, we present in the following some specific phenomena for both written and spoken Arabic:

- The word order in an utterance: Arabic speech does not satisfy any order rule since the order is totally variable.
- An Arabic word could represent a sentence in English. For example, the word « سنتحدّاهم » can express the sentence « we will challenge them ». Thus, word segmentation is necessary and difficult at the same time.
- The agglutination of coordinating conjunctions with the words also causes segmentation and tokenization problems.

3. Brief Overview of Automatic Speech Understanding Approaches

The speech understanding tries to assign the semantic representations of utterance. As mentioned in section 2, this analysis faces several problems. Different approaches have been proposed in the literature, and tested in various application contexts to solve these problems. In this section, we list the main approaches that have been proposed for the speech understanding. We distinguish three main approaches: the syntax guided approach, the semantic guided approach and the mixed approach.

3.1. Syntax Guided Approach

In this approach known as the approach guided by syntactic "ilots", a partial syntactic analysis is first planned. Indeed, in automatic speech analysis, it appears that a deep parsing could decrease the performance in the presence of unknown words, specific oral constructions, recognition errors and phenomena due to the spontaneity of the interaction.

Several known systems are based on this approach, such as the ROMUS system (Goulian, 2002 ; Goulian et al., 2003)) and the LOGUS system (Villaneau et al., 2004 ; Villaneau, 2007). These systems are dedicated to speech understanding applied to the domain of tourist information. Both systems are based on merely similar general architectures (segmentation into chunks and global analysis of dependencies). As ROMUS, LOGUS uses an incremental analysis strategy based on a partial analysis. This system uses the Lambda-calculation for the semantic representation of the oral utterance.

3.2. Semantic Guided Approach

This approach, known as a selective approach, benefits from the finalized nature of the dialogue. In this approach, understanding is limited to the search of the useful meaning of the utterance. It is based on the identification of the key sequences from which a predefined semantic structure will be instantiated. So, the idea is to analyze only the utterance parts that are considered to be relevant. We present, in what follows, few systems based on this approach.

The PHOENIX system (Ward, 1994) provides to users information about air transport. It includes a flexible semantic analyzer based on case grammars and compiled in a set of recursive transition networks. The analyzer provides as an output, a semantic representation of the utterance in frame forms. The understanding principle consists in detecting the segments corresponding to the attribute/value couples and generating the frame. In a pre-treatment step, simple mechanisms are applied to repetition detection and correction and to speech repairs. However, the out-of-vocabulary words are ignored. The analyzer keeps track of the dialogue and can solve the ellipses, anaphora and other indirect references.

For Arabic speech understanding, we cite the semantic decoder of Zouaghi (2007) applied to the railway domain (Zouaghi et al., 2007). This decoder uses a probabilistic semantic grammar that allows taking into account several contextual information at the same time. The semantic representation of the utterance is a Sense Representation Structure (SRS). The semantic decoding of utterances is based on a semantic analysis and considers only significant elements for the application (Zouaghi et al., 2006). Each significant word is represented by a set of semantic features noted Tse and a set of syntactic features noted Tsy where:

- Tse = {domain, semantic class, micro-semantic feature}
- Tsy = {gender, number, nature}

The analysis is based on a probabilistic language model that contributes to the selection of Tse to assign to the utterance words, and on a semantic lexicon that describes the meaning of each word through a set of Tse and a set of Tsy.

The selective approach seems to be effective when the domain is very limited and the semantic ambiguities are reduced.

In addition, it has certain robustness in facing difficulties caused by the spontaneity of oral utterances, the ungrammaticality of spoken language and the presence of misrecognized or unknown words. This robustness is obtained by the selective nature of this approach. Indeed, the utterance is not analyzed as a whole but only the parts expressing information relating to the application domain are considered. Otherwise, the understanding systems based on the selective approach and using the frame grammar formalism, have proven their performance in several domains. The main task for the development of such systems lies in the definition of the concepts and reference words. It is an important task, which is difficult but feasible for restricted domains.

3.3. Mixed Approach

This approach represents a combination of the two approaches describes in the previous sections (syntax guided approach and semantic guided approach). In this approach, a complete parsing is first considered. In case of failure of this analysis, semantic techniques are then used.

The TINA system (Seneff, 1989 ; Seneff, 1992) is based on a mixed approach. It provides information about airline in the United States. The understanding module of TINA is based on a context-free grammar. It uses a top-down analysis based on syntactic rules in addition to semantic constraints. The rules, which are those of a context-free grammar, are obtained directly from the analysis of the types of utterances. The utterance is analyzed completely from its syntax form. The grammar is automatically transformed into probabilistic automaton allowing advantaging the most frequent constructions. The inadequacy of the complete analysis to spontaneous speech, has led to the integration of a strategy of robust analysis using extended charts. This analysis allows the outstanding of correctly analyzed fragments (i.e. utterance parts) in case of failure of the complete analysis. The result of analysis is a tree. While several solutions are possible, the solution that considers more words is retained. The nodes of the tree correspond to the categories that can be syntactic or semantic.

4. Our Method

Our method of understanding is based on the selective approach. The main idea of our method is to guide the analysis to the local syntactic cores of the utterance. Thus, the syntax plays a minor role in the understanding process and does not require a deep linguistic knowledge. Moreover, and because of the ungrammatical nature of oral utterances, semantic extraction cannot totally rely on the syntactic analysis, as it is used for written texts. This extraction must be limited to the parts expressing the query information.

For all these reasons, we opted for the choice of frame grammar formalism which allows building, from an utterance, a semantic representation in the form of one or more frames. This formalism is one of the few formalisms that allows the treatment of ungrammatical utterances. In addition, this formalism offers a model of the deep structure of an utterance in which the semantics plays a main role, without excluding the syntactic constraints.

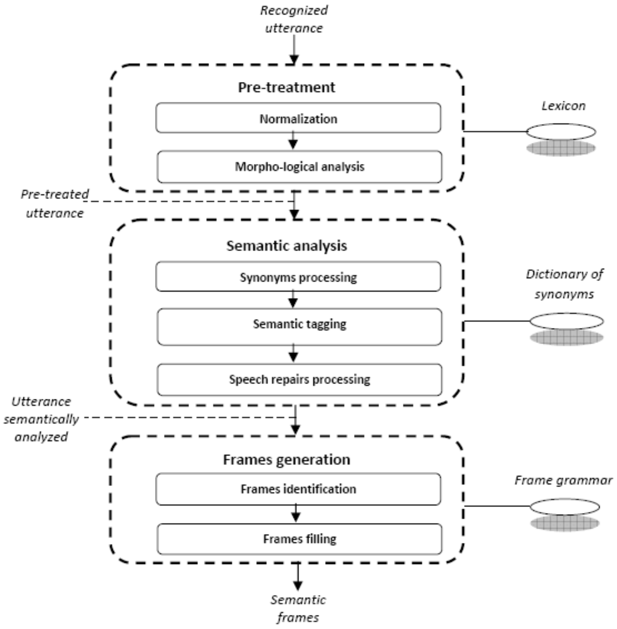


Figure 1: Main steps of our method.

Our method of utterance understanding consists of three major steps: a step of pre-treatment that includes the normalization of the utterance and its morpho-logical analysis; a step of semantic analysis that assigns semantic tags to each lexical unit of query; and a step of frame generation that identifies and fills the semantic frames of the utterance. Figure 1 shows the steps of our method.

4.1. Pre-treatment

The specific problems of Arabic language processing which are described in section 2 produce rigid structures that are difficult to handle. These problems are due to the spontaneity of interaction (repetitions, hesitations, speech repairs, etc). They are also due to several varieties of markers, reference words and to the specific vocabulary even if it is a restricted domain vocabulary. This situation leads to ambiguities which may cause analysis errors. Therefore, in our approach, each utterance undergoes a pre-treatment to normalize it and to determine its corresponding morpho-logical tags in order to facilitate the ulterior steps. The aim is to reduce the number of reference words and semantic cases that guide the choice of semantic frames.

- **The normalization**: it converts the transcribed utterance in a suitable form for ulterior analysis. Thus, the text numbers are converted into standard numbers (e.g. « خمسة عشر » « fifteen » becomes 15). These numbers concerns several semantic cases: the number of places to reserve, the fares, the departure and arrival dates (day, month and year), the departure and arrival times (hours and minutes). This normalization is very useful to facilitate the access to the database in order to extract information.
- **The morpho-logical analysis:** it first replaces the words of the utterance by their canonical forms[1]. Indeed, the reference words and the case markers of the frames as well as the lexicon entries are used in their canonical forms. This reduces the size of the frames and the lexicon since they contain the canonical forms rather than the different forms of the words with their derivatives. Second, if two or more consecutive words of the utterance are identical (a repetition case), we retain only one word. The out-of-vocabulary words (words that do not belong to the lexicon) are eliminated.

Let's take as an example the following utterance:

(1) « أريد معرفة سعر الرحلة من سوسة صفاقس نحو تونس تونس في الخامس عشر من ماي »

« I would like to know the travel cost from Sousse Sfax to Tunis Tunis on May fifteen »

The pre-treatment of this utterance first eliminates the out-of-vocabulary words « أريد معرفة » « I would like to know ». Also it eliminates the word « تونس » « Tunis » since it is repeated twice and replaces the word « الرحلة » « the travel » by its canonical form « رحلة » « travel » and the grouping words « الخامس عشر » « fifteen » with the

[1] The canonical form represents the root for verbs and the singular masculine not-determined form for the nouns and adjectives (Belguith Hadrich & Chaaben, 2006).

number 15. Thus, this utterance is transformed after pre-treatment into:

« سعر رحلة من سوسة صفاقس نحو تونس في 15 من ماي » **(2)**

« Travel cost from Sousse Sfax to Tunis on May 15 »

4.2. Semantic Analysis

The purpose of this step of semantic analysis is to generate a sequence of the couples < *Semantic Tag , Useful Unit* > of the pre-treated utterance. Thus, the reference words and case markers of the pre-treated utterance are replaced by their synonyms. Then, the utterance undergoes a semantic tagging and a speech repairs processing.

The replacement of the words by their synonyms concerns both markers and reference words. For example, the words « سعر », « كلفة », « تعريفة » and « مبلغ » are synonyms of the word « ثمن » « cost » and refer to the *Travel_Tariff* frame. If a word of this list occurs in an utterance, it is replaced by the word « ثمن » « cost ». Thus, the *Travel_Tariff* frame is lightened because it contains only the word « ثمن » « cost » as reference word and not the entire list of synonyms.

The semantic tagging assigns to each useful word of the utterance a semantic tag. As an example, the tag « *Depart_City_Mark* » is assigned to the preposition « من » « from ».

The speech repair processing consists to detect and to correct the utterance units that express self-corrections. Thus, if two consecutive words of an utterance have the same semantic tag, the first word is considered the wrong one, and will be eliminated. For example, in the utterance (2), the two successive words « سوسة » « Sousse » and « صفاقس » « Sfax » have the same semantic tag « *Depart_City* ». It is therefore a case of self-correction of the departure city. So, the first city « سوسة » « Sousse » is eliminated however the second city « صفاقس » « Sfax » is kept and considered as the departure city of this utterance.

The semantic analysis of utterance (2) gives the following sequence of words with the corresponding semantic tags:

< Depart_City_Mark,من > < Travel,سفر > < Tariff,ثمن > **(3)**
< Arrival_City_Mark,الى > < Depart_City,صفاقس >
< Depart_Day,15 > < Arrival_City,تونس >
< Depart_Month,ماي >

4.3. Frame Generation

The purpose of frame generation is the extraction of information from the utterance resulting from the two previous steps to produce the corresponding semantic frames. Notice that a frame is made up of several semantic cases and reference words (see simplified *Travel_Tariff* frame of figure 2). The main objective of this step of frame generation is to identify the semantic frames of the utterance by the reference words, and to fill these frames thanks to the markers and the semantic tags.

The frame identification is based on the list of reference words. It is done by scanning all semantic frames of grammar, and by calculating for each of them a score of detection of reference words. This score represents the number of reference words that are common to the utterance and the frame. Thus, the frame which has the highest score is selected. If several frames have the same score, they will all be retained.

The filling process of the selected frames is based on the case markers and the semantic tags of the utterance words. Indeed, this process consists in instantiating the value of each *Semantic Case / Value* couple of the frame by information (units having meaning) contained in the utterance and which are identified by the case markers or the semantic tags.

```
<Travel_Tariff frame>
    <Reference words> {ثمن، ثمن+سفر} </Reference words>
    <Depart_City> {...صفاقس، سوسة}[من] </Depart_City>
    <Arrival_City> {...صفاقس، سوسة} [الى] </Arrival_City>
    <Ticket_Type> {ذهاب، ذهاب وإياب}</Ticket_Type>
    <Train_Class> {ثانية، أولى، رفاهة}</Train_Class>
    <Depart_Day> {...2 ،1...اثنين، ثلاثاء} </Depart_Day>
    <Depart_Month> {...جانفي، فيفري} </Depart_Month>
    <Depart_Hour>{...2 ،1} [ساعة] </Depart_Hour>
    <Depart_Minute> [دقيقة] {...2 ،1} </Depart_Minute>
</Travel_Tariff frame>
```

Figure 2: Simplified *Travel_Tariff* frame.

For sequence (3), two candidate frames are identified: the *Travel_Tariff* frame and the *Travel_Schedule* frame. The *Travel_Tariff* frame is identified by two reference words « ثمن » « cost » and « ثمن+سفر » « cost+travel ». As a result, it score is 2. The *Travel_Schedule* frame is identified by only one reference word « سفر » « travel » so it scores is 1. Thus, the retained frame is the *Travel_Tariff* frame since it has the highest score.

The following figure shows the *Travel_Tariff* frame of figure 2 which is generated by using information from sequence (3).

```
<Travel_Tariff frame>
    <Depart_City> صفاقس </Depart_City>
    <Arrival_City> تونس </Arrival_City>
    <Ticket_Type> $[2] </Ticket_Type>
    <Train_Class> $ </Train_Class>
    <Depart_Day> 15 </Depart_Day>
    <Depart_Month> ماي </Depart_Month>
    <Depart_Hour> $ </Depart_Hour>
    <Depart_Minute> $ </Depart_Minute>
</Travel_Tariff frame>
```

Figure 3: Simplified *Travel_Tariff* frame corresponding to the sequence (3).

5. Presentation of SARF System

In this section, we present our SARF system (Arabic vocal server of railway information). SARF is an interactive Arabic vocal server that offers users access in oral modern standard Arabic to Tunisian railway information.

[2] The $ symbol indicates that the semantic case is not instantiated.

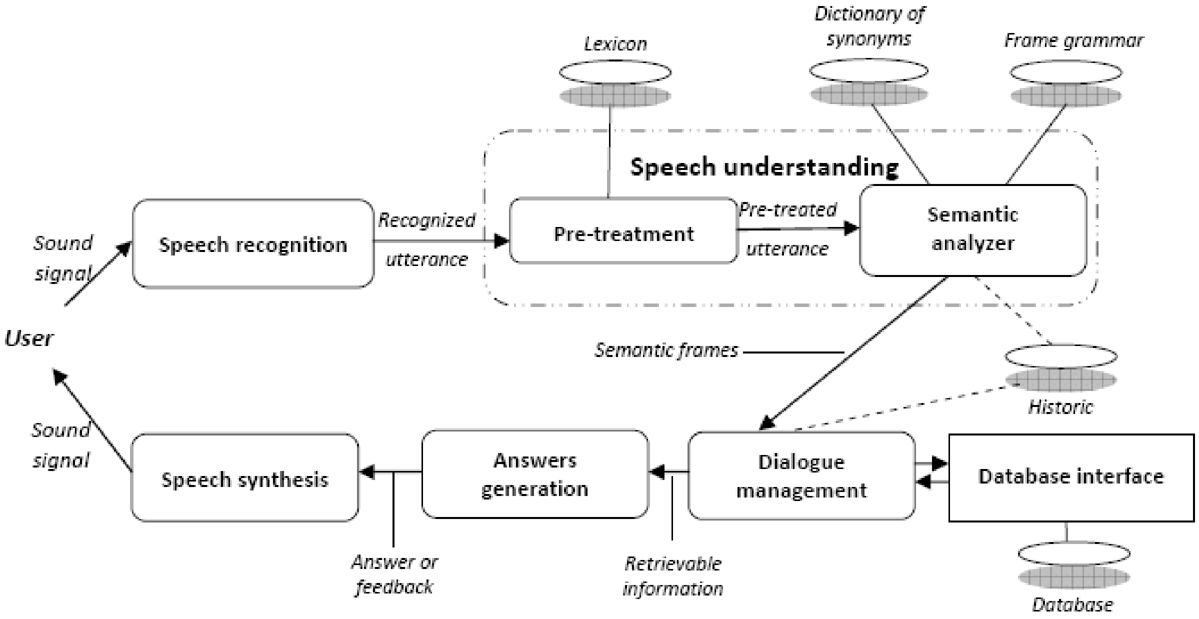


Figure 4: The general architecture of SARF system.

It is based on the frame grammar formalism for oral utterances understanding and applies the method that we have proposed in section 4.

The SARF system is composed of five modules namely, the speech recognition module, the speech understanding module, the dialogue management module, the answers generator module and the speech synthesis module. Figure 4 shows the general architecture of our SARF system.

SARF integrates the techniques of speech recognition, understanding, dialogue management and speech synthesis. These techniques make possible the extraction of the meaning of an utterance pronounced by the user, in order to provide him with the required information.

From the signal emitted by the speaker, the speech recognition module generates one or several lists of words that are supposed to correspond to the source utterance. The understanding module provides for the dialogue manager one or more semantic representations of the transcribed utterance. The dialogue manager assures the interface with the database and suggests answers, or demands additional information to the user. The answers generator module assures the transformation of the answers to Arabic sentence understandable by the user. The speech synthesis module allows their transmission into sound signal.

In what follows, we focus on SARF understanding module. Note that for the speech recognition module and the speech synthesis module, we plan to use commercialized systems.

The understanding module is made up of two sub-modules namely **(i)** the pre-treatment which assures the normalization and the morpho-logical analysis of oral utterances and **(ii)** the semantic analyzer which allows the utterance semantic tagging and the semantic frame generating.

The frame grammar of SARF understanding module includes six semantic frames namely, *Travel_Tariff, Travel_Schedule, Journey_Time, Ticket_Reservation, Train_Itinerary and Train_Type*. Each frame contains reference words and semantic cases related to our application domain.

In order to reduce the number of reference words and the number of case markers of the frames, SARF understanding module uses a dictionary of synonyms. Furthermore, to reduce the size of the lexicon, each word in the lexicon is reduced to its canonical form. Thus, SARF uses the morphological analyzer MORPH 2 (Belguith Hadrich & Chaaben, 2006) to determine the canonical forms.

The SARF understanding module is implemented with the *JBuilder 10* environment using JAVA programming language. The lexicon, the dictionary of synonyms and the frame grammar are stored in XML files to ensure their portability and flexibility.

6. Evaluation of SARF Understanding Module

Since Arabic language resources are very rare and merely unavailable, we were obliged to build our own evaluation corpus using the technique of Wizard of Oz. Thus, we have used scenarios dealing with information on Tunisian railways. All queries were recorded, and then manually transcribed according to standards of transcription in XML files, and tagged in accordance with the standards proposed by the ARPA community. We distinguish three types of queries: context independent queries (query type A), context dependent queries (type D) and aberrant queries (query type X).

This corpus includes 1003 queries representing 12321 words. The queries contain different types of information and several difficulties of spontaneous speech (repairs, repetitions, etc). The evaluation corpus reflects different levels of users, as well as various types of utterances.

The evaluation of SARF understanding module has shown that this module produced 186 errors where 67 errors are due to the failure of MORPH 2 to find the appropriate canonical forms. The recall, precision and F-Measure rates are respectively 73.00%, 70.62% and 71.79%. Note that the average time of execution of an utterance is equal to 0,279 second. The error rate of 18.54% is mainly due to the presence of truncated words and to the misrecognized words. These two phenomena are not handled by our system yet. Let's take for example the utterance (4), which is not correctly analyzed by our system.

(4) « ما هو ثمن تذكرة القطار **العاد** »

« How much does the ticket of **norm**[3] train cost »

In this utterance, the user does not pronounce correctly the long vowel of the word « العادي » « normal ». Therefore, this word has been truncated and transcribed as the word « العاد » « norm ». Since the truncated word does not figure in our lexicon, it is eliminated by our system. Consequently, there will be errors in filling the semantic frame corresponding to the utterance (4).

6. Conclusion and Perspectives

In this paper, we have proposed a method for spontaneous Arabic speech understanding in a context of human-machine spoken dialogue. This method is part of the semantic guided approach and is based on the frame grammar formalism for the semantic representation of oral utterances.

We have also presented the SARF system which is an interactive Arabic vocal server that allows to users access, in oral modern standard Arabic, to Tunisian railway information. The SARF understanding module is based on the proposed method. The assessment results are very encouraging even if SARF understanding module, in its current state, does not handle yet some problems such as: truncated words and misrecognized words. Indeed, we obtained a rate of 71.79% for the F-Measure.

As perspectives, we plan to evaluate SARF on a larger corpus. Also, we plan to study and solve the problems related to truncated words and misrecognized words.

References

Belguith Hadrich, L. & Chaaben, N. (2006). Analyse et désambigüisation morphologiques de textes arabes non voyellés. TALN'06, Leuven, Belgique.

Bruce, R. (1975). Cases Systems for Natural Languages. Artificial Intelligence, Volume 6, pp. 327-360.

Fillmore, C.J. (1968). The case for case. In Universals in Linguistic Theory, E. Bach and R.T. Harms (Eds.), Holt Rinehart, New York.

Fillmore, C.J. (1977). The case for case re-opened. In Syntax and Semantics 8, P. Cole and J.M. Saddock (Eds.), Academic Press, New York.

Goulian, J. (2002). Stratégie d'analyse détaillée pour la compréhension automatique robuste de la parole. PhD thesis, Université de Bretagne Sud, Vannes, France.

Goulian, J., Antoine, J.-Y. & Poirier, F. (2003). How NLP techniques can improve speech understanding: ROMUS – a Robust Chunk based Message Understanding System Using Link Grammars. EUROSPEECH'03, Geneva.

Minker, W. & Bennacef, S. (2005). Speech and Human-Machine Dialog. Computational Linguistics, Volume 31, Number 1, pp. 157–158.

Ramsay, A. & Mansour, H. (2008). Towards including prosody in a text-to-speech system for modern standard Arabic. Computer Speech and Language, Volume 22, Issue 1, pp. 84-103.

Satori, H., Harti, M. & Chenfour, N. (2007a). Arabic Speech Recognition System based on CMUSphinx. ISCIII'07, Agadir, Maroc.

Satori, H., Harti, M. & Chenfour, N. (2007b). Introduction to Arabic Speech Recognition Using CMU SphinxSystem. International Journal of Computer Science.

Seneff, S. (1989). TINA: A probabilistic syntactic parser for speech understanding systems. In Proceedings of the Speech and Natural Language Workshop, Philadelphia, USA.

Seneff, S. (1992). TINA: a Natural Language System for Spoken Language Applications. Computational Linguistics, Volume 18, Number 1, pp. 61–86.

Van Schooten, B., Rosset, S., Galibert, O., Max, A., Op Den Akker, R. & Illouz, G. (2007). Handling speech input in the Ritel QA dialogue system. Proceedings of INTERSPEECH'07, Antwerp. Belgium.

Villaneau, J. (2007). Une expérience de compréhension en contexte de dialogue avec le système LOGUS, approche logique de la compréhension de la langue orale. TALN'07, Toulouse, France.

Villaneau, J., Ridoux, O. & Antoine, J.-Y. (2004). LOGUS: compréhension de l'oral spontané: présentation et évaluation des bases formelles de LOGUS. RSTI-RIA'04, Revue d'Intelligence Artificielle, Volume 18, Number 5-6, pp. 709-742.

Ward, W. (1994). Extracting Information in Spontaneous Speech. In Proceedings of International Conference of Speech and Language Processing, ICSLP'94, Yokohama.

Zouaghi, A., Zrigui, M. & Ben Ahmed, M. (2006). L'influence du contexte sur la compréhension de la parole arabe spontanée. TALN'06, Leuven, Belgique.

Zouaghi, A., Zrigui, M. & Ben Ahmed, M. (2007). Évaluation des performances d'un modèle de langage stochastique pour la compréhension de la parole arabe spontanée. TALN'07, Toulouse, France.

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, J.T. & Hetherington, L. (2000). JUPITER: A telephone-based conversational interface for weather information. IEEE Trans., on Speech and Audio Processing, Volume 8, Number 1.

[3] The word « normal » has been truncated.

Methods for porting NL-based restricted e-commerce systems into other languages

[*]Najeh Hajlaoui, [**]Daoud Maher Daoud, [*]Christian Boitet

[*]GETALP, LIG, Université Joseph Fourier
385 rue de la Bibliothèque, BP n° 53
38041 Grenoble, cedex 9, France
Najeh.Hajlaoui@imag.fr,Christian.Boitet@imag.fr

[**]Amman University
PoBox 141009, Zip Code 11814
Amman Jordan
Daoud@ammanu.edu.jo, Daoud.Daoud@imag.fr

Abstract

Multilingualizing systems handling content is an important but difficult problem. As a manifestation of this difficulty, very few multilingual services are available today. The process of multilingualization depends on the translational situation: types and level of possible accesses, available resources, and linguistic competences of participants involved in the multilingualization of an application. Several strategies of multilingualization are then possible (by translation, by internal or external localization etc.). We present a real case of linguistic porting (from Arabic to French) of an e-commerce application deployed in Jordan, using spontaneous SMS in Arabic for buying and selling second-hand cars. Despite the distance between Arabic and French, the localization methods used give good results because of the proximity of the two sublanguages of Arabic and French in this restricted domain.

Introduction

Methods for multilingualizing e-commerce services based on content extraction from spontaneous texts depends on two aspects of the translational situation:

- the level of access to resources of the initial application. Four cases are possible: complete access to the source code, access limited to the internal representation, access limited to the dictionary, and no access.
- the linguistic qualification level of the persons involved in the process (level of knowledge of the source language, competence in NLP) and the resources (corpora, dictionaries) available for the new language (s), in particular for the "sublanguages" at hand.

We first discuss the requirement in localizing the Content Extractor (CE, or content extraction module) of such an application, and an analysis of the possible methods. We then present a case study, the linguistic porting (from Arabic into French) of CATS, an application to which we have complete access, so that several multilingualization strategies are possible. In the next section, we present an "external" localization strategy which requires only access to the internal representation. Then, we evaluate this method by comparing it with the results produced by the original monolingual system. We also compare it with the "internal" localization method, which consists in adapting the existing content extractor, and has been described in another paper (Hajlaoui 2007).

1 Multilingualizing NLP-based services

We concentrate on NLP-based systems that perform specific tasks in restricted domains, Figure 1 shows the general structure of these systems. Examples of such applications and services are: categorization of various documents such as AFP (Agence France Presse) reports or customer messages on a SAS (Service After Sale) server, and information extraction to feed or consult a database (e.g. small ads, FAQ, automated hotlines).

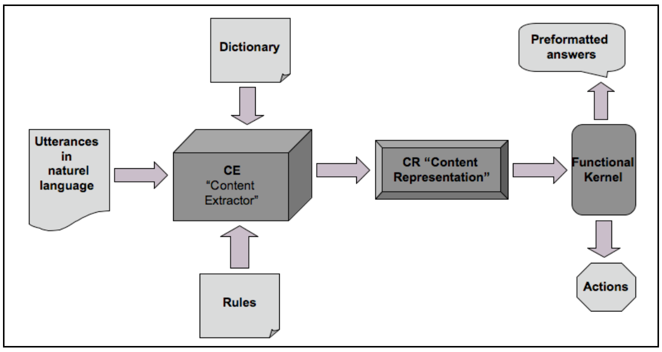


Figure 1: general structure of NLP-based systems

1.1 Benefits and necessity of NL interfaces

The main benefits of building systems based on the processing of spontaneous unedited text are:

- the naturalness of interaction,
- the ability of expressing complex expressions,
- the possibility (more recently) of designing and building domain-focused services based on a domain-specific thesaurus or ontology.

Integrating a module processing small spontaneous task-related texts (ads, messages…) seems to be the only answer to the growing ergonomic needs, especially for e-commerce applications.

The study of the current scene shows that the deployed or operational e-commerce NL interface systems are rare and most of them are only prototypes.

Furthermore, when we find an application, we get very few information about its internal procedure and how its multilinguization has been done or envisaged.

Among the few we found are Pertinence Summarizer (Lehman 1996), a system of automatic summarization of multilingual texts, Amilcare (Ciravegna 2001), an adaptive system of information extraction, NLSA "Natural Language Sales Assistant", a dialogue-based system through the Web deployed by IBM, and CATS "Classifieds Ads Transaction System" (Daoud 2006), a

Arabic-based SMS system for handling classified ads related to buying and selling cars and real estate.

1.2 Sublanguage and content extraction

The reasons why handling spontaneous text is difficult are similar to those encountered in speech processing: non standard grammar (more or less oral), errors (spelling, typing errors), use of some typographical conventions (SMS-specific abbreviations, smileys). Often enough, classified ads or alarms/warnings (road traffic, natural disasters) form a sublanguage which is relatively far from the general language.

The consequence is that it is not possible to use the "analysis approach" and associated tools, developed for written general and clean language, even by specializing them. Rather, it seems necessary to adopt a content extraction approach, using the particularities of the sublanguage at hand.

2 Possible approaches to "porting"

2.1 Common need: a corpus in L2

The primary data needed for porting an application treating spontaneous texts in language L1 to language L2 is a representative corpus in L2 of the same type of data… but that is almost never available. It is hence generally necessary to create one by translation, or by simulation or imagination, and post-edition.

For example, the *Real Estate* part of CATS treats SMS concerning Amman. To *localize* it to French, one should handle cities in France, Belgium, Switzerland, Canada, or Africa etc., and hence one should develop a corpus covering the various corresponding sublanguages. That would be a too expensive process. Because of that problem, we limit our ambition to *porting* and not localizing applications. In the case of CATS, porting to French means that French users could use CATS for sending French SMS in the *same* situation (real estate in Amman), so that a suitable corpus could then be created by translating and post-editing the corpus of available SMS from Arabic to French.

The “good size” of the initial L2 corpus depends on the strategy used. If we adapt an existing CE, the translation of the L1 corpus used for initial development or for regular testing should suffice. If we develop an L2-L1 MT system, we might need a much larger corpus. However, a recent experiment seems to indicate that a SMT usable by the CE to work well can be developed with a rather small corpus, and a complete dictionary.

2.2 Translation

If there is no access to the code, dictionary, and internal content representation of the target application, the only possible approach to localize it from L1 to L2 is to develop an MT system to automatically translate its (spontaneous) inputs from L2 into L1.

This approach might look applicable and straightforward. However, from the practical perspective it is not. As a matter of fact, for restricted domains where we have special sublanguage with specific terminology, traditional MT is not useful.

2.3 Porting or adapting a CE

Each considered application `App` uses a specific CRL (content representation language), say `CRL_app`. Several types of content representation are used, such as property lists (<attribute, value> pairs), typed features structures, logical expressions (Prolog), logico-functional expressions, objects (classes (methods, attributes), instances). Deriving a CE from one application for another one is difficult, even in the same language, because one must guarantee a minimum level of quality, correctness and completeness of the extracted content, as well as the relevance and linguistic adequacy of the produced answers.

Localization at the level of content extraction can be achieved by "internal" porting or "external" adaptation.

2.3.1 Internal CE localization

The first possibility consists in adapting the CE of the application from L1 to L2; but that is viable only if

- the developers agree to open their code and tools,
- the code and tools are relatively easy to understand,
- the resources are not too heavy to create (in particular the dictionary).

That method requires of course training the localization team with the tools and methods used.

Under these conditions, adaptation can be done at a very reasonable cost, and further maintenance (to "follow" the drift of the sublanguage) can later be done cheaply.

2.3.2 External CE localization

The second solution consists in adapting an available CE for L2 to the sublanguage at hand, and to translate its results into the target CRL_app.

For a company wanting to offer multilingualization services, it would indeed be an ideal situation to have a generic CE, and to adapt it to each situation (language, sublanguage, domain, CRL, task, other constraints). However, there are still no known generic CEs of that power, and not even generic CEs for particular languages, so that this approach cannot be considered at present.

A third approach is then to adapt an existing content extractor, developed for L2 and a different domain/task, or for another language and the same domain/task.

We have previously experimented the first method (Hajlaoui 2007) by porting the *Cats* part of CATS from Arabic to French: for that, we adapted its native Arabic CE, written in EnCo, by translating its dictionary, and modifying a few analysis rules.

We also tried the third method, on exactly the same task, and report on that experiment in the following section.

3 Case study: external CE adaptation

3.1 Presentation of CATS

CATS (Classifieds Ads Transaction System) is a platform for buying and selling goods (cars, real estate…) based on the use of Arabic SMS and created by the second author (Daoud 2006). It is deployed by Fastlink (the largest mobile operator in Jordan). It is a C2C based e-commerce system that uses content extraction technology based on sublanguage analysis and knowledge representation to enable SMS users to post and search for classified ads in Arabic. It has two main functionalities: the submission for selling items and the answering of users' queries through interaction in spontaneous natural language. The system receives an entry in full text without a pre-specified layout, recognizes the various relevant entries, and produces a knowledge representation for further processing. We have two types of users' requests:

- "Sell" post: in which the user is a potential seller.
- "Looking for" post: in which the user is a potential buyer.

Table 1 shows some examples of the car domain, for which we are interested in a first time.

مطلوب سياره هونداي موديل 97 والسعر مابين 3500 الى 3750	Looking for Honda, model 97, price between 3500 and 3750
مطلوب سياره سبور	Looking for sport car
اريد سياره مرسيدس موديل 82 لون ابيض	I want Mercedes car model 82 white color
سياره اوبل فكترا للبيع موديل 2003 فل اوبشن	Opel Vectra car for sale year2000 full option
للبيع سيارة بي ام دبليو 520 لون زيتي فحص كامل م 89 فل عدا الفتحه مرخصه بحال ممتازه بسعر 8500	For sale BMW 520 color dark green full check year 89 full except sunroof licensed in a good condition with a price 8500.
أوبل أسترا ستيشن لون أحمر (بورفتحه سنترزجاج ومريات كهرباء) فحص للبيع .	Opel Astra station color red (power sunroof Center Electrical windows and mirrors check for sale
عندي سياره لاند روفر بدي ابيعها.	I have a Land Rover car I want to sell it
مطلوب سيارة بيجو406	Wanted a Peugeot 406 car
بحاجه لسياره لا تزيد عن 2000 دينار بحاله جيده واقتصاديه في البنزين	In need for a car not more than 2000 dinar in good condition and economical in fuel.
شراء سيارة.	Buying a car
اريدبيع سياره دايو ليمنز موديل 92 فحص كامل فل اوبشن	I want to sell a Daewoo Lemens car year 92 full check full option

Table 1: examples in the cars domain

The overall structure of the CATS reflects both the corpus analysis and the adopted knowledge representation. The CATS system consists of a content extraction (CE) component and a query manager QM component.

The CE component receives SMS text and decodes it into the corresponding knowledge representation CRL-CATS using a domain-specific lexicon. The system is able to extract knowledge from both types of messages.

The QM component takes the KR and converts it into SQL statements. It also issues the SQL statements (query or insert), and checks, validates and formats the results. It also handles situations where no answer found.

One important aspect of this design is that both questions and postings (documents) are processed by the same engine, using the same knowledge representation, leading to accurate matching of questions with answers.

CE is written in EnCo (Uchida and Zhu 1999) and uses a lexicon specific to the domain and a grammar specific to the sublangage.

```
 ;للبيع رينو ميجان م 2000

;Selling Renault Megane m 2000

[S]
sal(saloon:00,sale:00)
mak(saloon:00,RENAULT(country<France,county
<europe):07)
mod(saloon:00,Megane(country<France,country
<europe,make<RENAULT):0C)
yea(saloon:00,2000:0K)
[/S]
```

Figure 2: a CRL-CATS representation

In CRL-CATS (Content Representation Language for CATS), a posted SMS is represented as a set of binary relations between objects. It is a kind of semantic graph with a UNL-like syntax. There are no variables, but the dictionary is used as a type lattice allowing specialization and generalization.

In the preceding example, there is one object, a car (`saloon`), with 4 properties. The first is `sal` (type of post, selling or buying, here `sale` for selling). The other properties are `mak` (make), with value `RENAULT (country<France,country<europe)`, `mod` (model), with value `Megane(country<France, country< europe,make<RENAULT)`, and `yea` (year), with value `2000`.

3.2 Necessity of an "initial" corpus

For all localization methods of the CATS system into French, the first thing to do was to collect or build a "French starting corpus", similar to that used by Daoud at the beginning of his project for Arabic. That was obviously necessary to study the syntactic form of the SMS to be treated in French, and to see also which lexical categories to expect. Initially, we used an Arabic corpus generated by CATS and translated it into "French spontaneous SMS", expected to be sent (in Jordan) by French-speaking people.

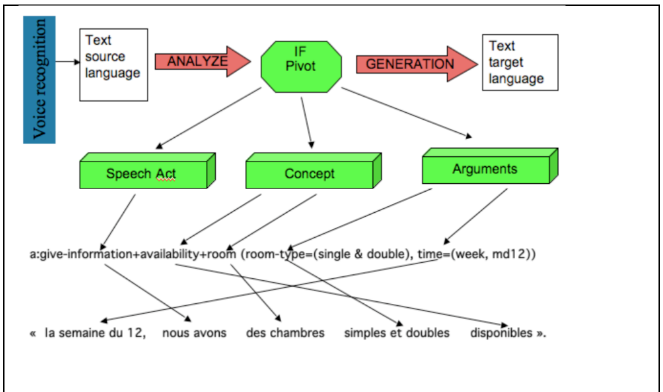
A rough translation produced by a non-French person is generally very different compared to a natural and functional translation produced by a French person, i.e. compared to what a French person would say in a spontaneous way in the same situation. We evaluated this translation difference between rough/literal and natural/functional translations by calculating the edit distance between two translations. The average distance is 21,88 (Hajlaoui 2006), for an average length of 66 characters.

In order to develop this corpus, we adopted the following technique: starting from the ads model constituted by 50 types of SMS revised and considered to be functional, we multiplied the number of these ads by forming different combinations of properties and values (make, model, year, colour, price…). For example, we replace a year by another (je cherche une voiture modèle 98) → je cherche une voiture modèle 99…) or a make by another (A vendre BMW rouge → A vendre PEUGEOT noire).

3.3 External localization

We adapted an existing French extractor of 31,918 lines written in Tcl/Tk by H.Blanchon for the Nespole! project (Blanchon 2004), and producing IF (Interchange Format) representations. The IF is a semantico-pragmatic pivot used for restricted domains. The second step was then to translate the IF expressions into CRL-cats graphs.

The IF is a semantico-pragmatic pivot used for restricted domains. Figure 3 shows the IF specification components: speech act, concept and arguments. In the beginning of this adaptation, we have the code of the second demonstrator, the paper and electronic version of the IF specification (version of 08-18-2002) and the CRL-CATS specification.


Voice recognition
Text source language
ANALYZE
IF Pivot
GENERATION
Text target language
Speech Act
Concept
Arguments
a:give-information+availability+room (room-type=(single & double), time=(week, md12))
« la semaine du 12, nous avons des chambres simples et doubles disponibles ».

;the week of 12, we have simple and double rooms available

Figure 3: content extractor for French in tourism domain (Blanchon 2004)

3.3.1 Content extraction method in Nespole!

Blanchon's CE uses a method based on finite-state transducers. Relevant sequences are described by regular expressions and attached actions incrementally consume the input and produce an IF expression.

We tried to understand and use the method used in the second module of Blanchon’s CE. As Figure 4 shows, the method used for the analysis (French to IF) has the following stages:

- Segmentation in SDU (Semantic Dialogue Units).
- Detection of the domain.
- Construction of speech acts prefix and instantiation of dependent arguments.
- Instantiation of arguments related to domain and management of subordinations.
- Complementation of speech acts.

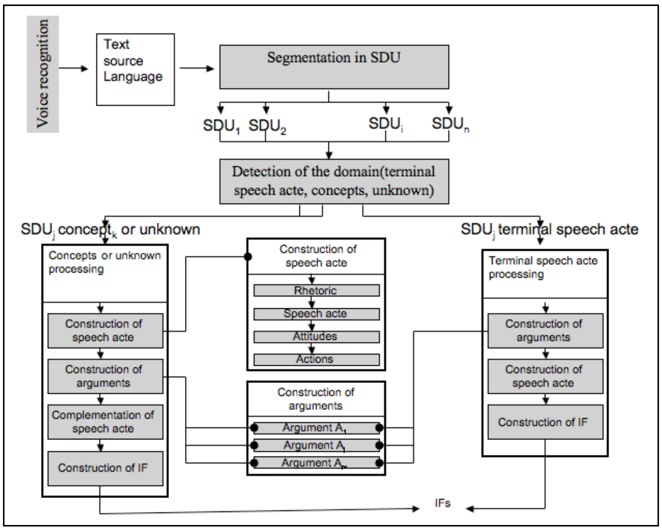

Voice recognition
Text source Language
Segmentation in SDU
SDU1 SDU2 SDUi SDUn
Detection of the domain(terminal speech acte, concepts, unknown)
SDUj conceptk or unknown
SDUj terminal speech acte
Concepts or unknown processing
Construction of speech acte
Construction of arguments
Complementation of speech acte
Construction of IF
Construction of speech acte
Rhetoric
Speech acte
Attitudes
Actions
Construction of arguments
Argument A1
Argument A2
Argument An
Terminal speech acte processing
Construction of arguments
Construction of speech acte
Construction of IF
IFs

Figure 4: structure of French analysis module into IF for the NESPOLE second demonstrator (Blanchon 2004)

3.3.2 IF and French-IF CE adaptation

We adapted the IF specification to the *Cars* domain. We also enriched it by adding new arguments like vehicle-motor-type, vehicle-hand, etc.

We added new actions, essentially the buying action e-buy and the selling action e-sell. We used the same stages to extract information about the *Cars* domain. We tried to eliminate the instructions which posed problem and/or which were not necessary to reduce the computing time. We added new instructions related the added arguments and actions.

Most of the work was done on the arguments instantiation stage related to the vehicle domain: we instantiate the vehicle specification vehicle-spec, as well as other less interesting arguments such as: theDistance, theLocation, theDuration, theDestination, theTime, thePrice…

A new VehicleSpec2If function allows search and construction of the arguments related to the vehicle concept. The only argument already programmed in Blanchon's CE was frenchvehicle, which can have values voiture, ski, camion, bus, train, avion… Likewise, the Argument2if function builds the IF associated values. Figure 5 is an example of result obtained after adaptation.

Input 1 = A vendre une grande voiture française BM 325 4 portes diesel bleue TBE première main assurance complète avec CT sans climatisation TB prix dernier mod

English translation : For sale a big French car BM 325 4 doors diesel blue VGS first hand insurance completed with CT without air-cond VG price last mod

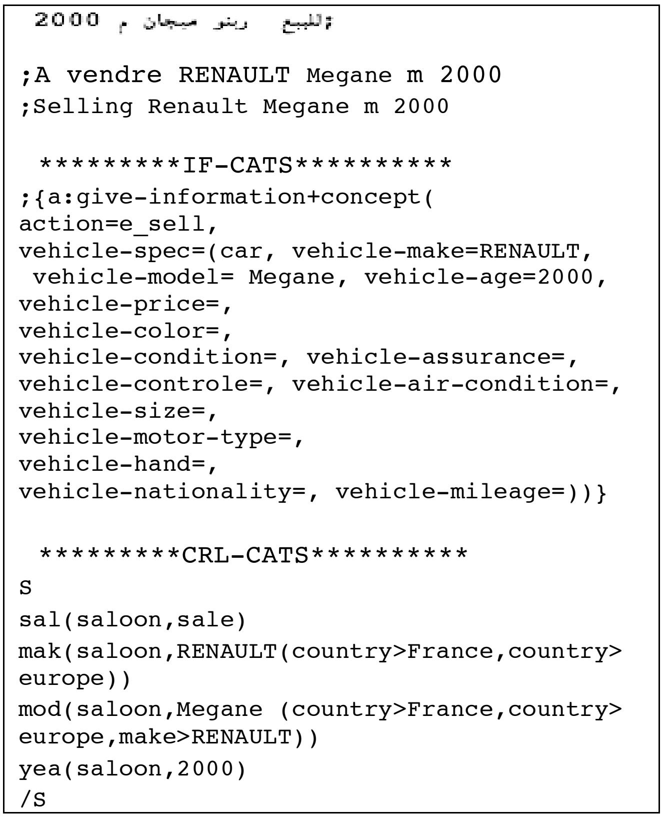
```
Output1 = {c:give-information+disposition+vehicle
(disposition=(desire, who=i),
 action=e_sell,
 vehicle-spec=
 (car, vehicle-make=BMW,     vehicle-model=325,
  vehicle-size=4_door,       vehicle-shape=big,
  vehicle-motor-type=diesel,
  vehicle-hand=first_hand,   vehicle-color=blue,
  vehicle-condition=good,    age-vehicle=new_mod,
  vehicle-assurance=insured,
  vehicle-controle=total_check,
  vehicle-air-condition=no_air_condition,
  vehicle-nationality=french,
  price-vehicle=good_price))}
```

Figure 5: result of content extraction on a French SMS

We call the obtained result “IF-CATS” (output1 in the example). We built a compiler, which analyses and transforms the IF-CATS expressions in CRL-CATS graphs by using an “IF-CRL” dictionary which facilitates the substitution of the arguments.

3.3.3 Compiler IF-CATS_CRL-CATS

We built a compiler, which analyses and transforms the IF-CATS expressions in CRL-CATS graphs by using an “IF-CRL” dictionary which facilitates the substitution of the arguments.
Figure 6 gives the same result of the CRL-CATS protected by the compiler. It is the same result as that of Figure 2. It shows that it is possible to obtain the same CRL-CATS format as that produced by the EnCo[1] tool, except for the symbols 00, 0J, 0R which are added by the EnCo tool.



```
 ;للبيع   رينو ميجان م 2000

;A vendre RENAULT Megane m 2000
;Selling Renault Megane m 2000

 *********IF-CATS**********
;{a:give-information+concept(
action=e_sell,
vehicle-spec=(car, vehicle-make=RENAULT,
 vehicle-model= Megane, vehicle-age=2000,
vehicle-price=,
vehicle-color=,
vehicle-condition=, vehicle-assurance=,
vehicle-controle=, vehicle-air-condition=,
vehicle-size=,
vehicle-motor-type=,
vehicle-hand=,
vehicle-nationality=, vehicle-mileage=))}

 *********CRL-CATS**********
S
sal(saloon,sale)
mak(saloon,RENAULT(country>France,country>
europe))
mod(saloon,Megane (country>France,country>
europe,make>RENAULT))
yea(saloon,2000)
/S
```

Figure 6: an IF-CATS_CRL-CATS compiler result

3.3.4 Results and evaluation

We translated manually the evaluation corpus used for the evaluation of CATS Arabic version (original). It contains 200 real SMS (100 SMS to buy + 100 SMS to sale) posted by real users in Jordan.
We spent 289 mn to translate the 200 Arabic SMS (2082 words is equivalent to 10 words/SMS, approximately 8 standard pages[2]) into a French translation or about 35 mn per page.
We spent 10 mn per standard page to pass from raw translation to functional translation.

We obtained 200 French SMS considered to be functional (1361 words, or about 6,8 words/SMS, approximately 5 standard pages).

We translated manually the corpus used for the evaluation of CATS Arabic version (original). We computed the recall R, the precision P and the F-measure F for each most important property (action “sale or buy”, “make”, “model”, “year”, “price”).

P = number of correct entities identified by the system/ total number of entities identified by the system;

R = number of correct entities identified by the system/ total number of entities identified by the human;

F = 2PR/(P+R)

Table 2 summarizes the results obtained and Table 3 shows details (Hajlaoui and Boitet 2007). Properties having numbers as values, like price and year, lower the percentage of porting by external adaptation, but the advantage is that that method requires only to access the internal representation of the application.

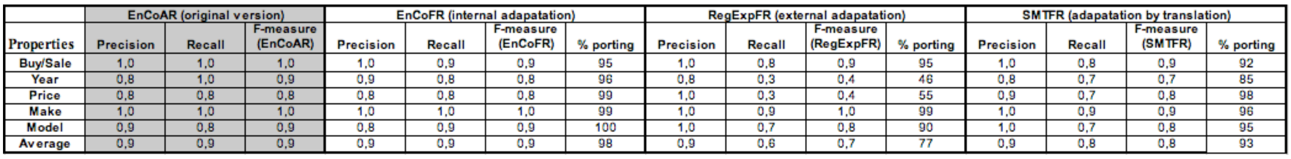
Porting by	minimum	average	maximum
internal adaptation	95%	98%	100%
external adaptation	46%	77%	99%
statistical translation	85%	93%	98%

Table 2: evaluation of three localization methods used for porting CATS_Cars from Arabic to French

Figure 7 allows to better visualize the comparison between the values of F-measure found for each version of the system.

[1] EnCo is a tool based on rules and dictionaries used for content extraction in original version of CATS system.

[2] Standard page = 250 words



Properties	EnCoAR (original version)			EnCoFR (internal adapatation)				RegExpFR (external adapatation)				SMTFR (adapatation by translation)			
	Precision	Recall	F-measure (EnCoAR)	Precision	Recall	F-measure (EnCoFR)	% porting	Precision	Recall	F-measure (RegExpFR)	% porting	Precision	Recall	F-measure (SMTFR)	% porting
Buy/Sale	1,0	1,0	1,0	1,0	0,9	0,9	95	1,0	0,8	0,9	95	1,0	0,8	0,9	92
Year	0,8	1,0	0,9	0,9	0,8	0,8	96	0,8	0,3	0,4	46	0,8	0,7	0,7	85
Price	0,8	0,8	0,8	0,8	0,8	0,8	99	1,0	0,3	0,4	55	0,9	0,7	0,8	98
Make	1,0	1,0	1,0	1,0	1,0	1,0	99	1,0	0,9	1,0	99	1,0	0,9	0,9	96
Model	0,9	0,8	0,9	0,8	0,9	0,9	100	1,0	0,7	0,8	90	1,0	0,7	0,8	95
Average	0,9	0,9	0,9	0,9	0,9	0,9	98	0,9	0,6	0,7	77	0,9	0,8	0,8	93

Table 3: comparison between result of content extraction

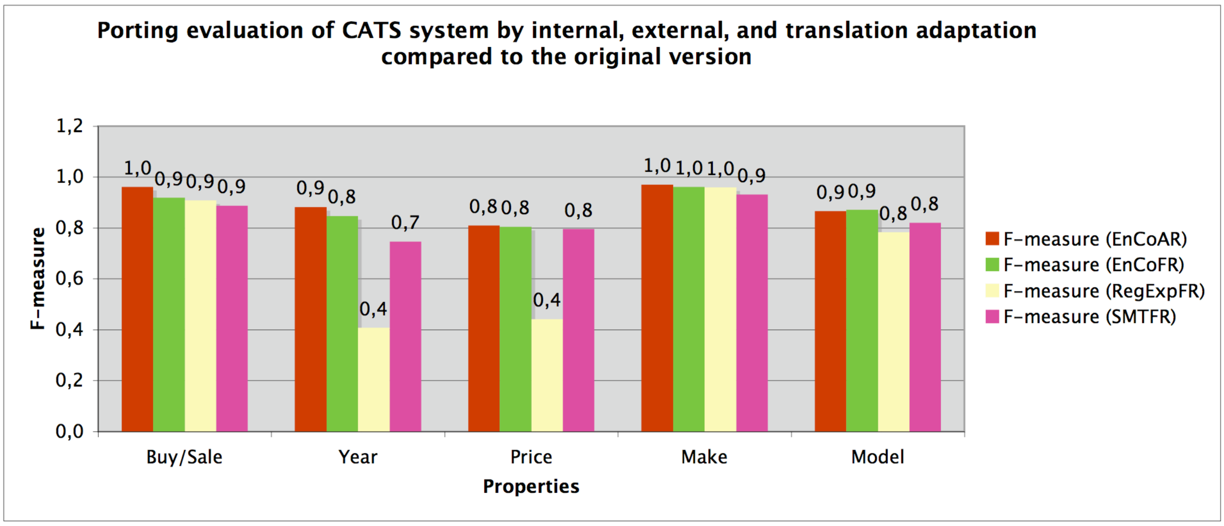


Figure 7: comparison between F-measures

Conclusion

We have presented several possible methods for "porting" applications based on handling the content of spontaneous NL messages in a "native" language L1 into another language, L2. In a previous paper, we described an experiment and an evaluation of the "internal" strategy, consisting in adapting the native CE (content extractor) to L2.

Here, we reported on another case study, in which we experimented the "external" strategy, consisting in adapting an existing CE for L2 to the domain and to the CRL (content representation language) of the target application. The evaluation was done similarly, on the same part of the CATS system, *Cars,* to port it from Arabic to French. Porting by translating messages from L2 to L1 (here from French to Arabic) has also been done, and its evaluation is also given.

An interesting conclusion of these experiments is that, in a real case of linguistic porting, all three localization methods used gave good results. The most likely reason for that seems to be that, although Arabic and English are quite distant as "general languages", their considered sublanguages are quite similar. That corroborates the analysis made by (Kittredge 1986).

Acknowledgments

We would like to thank our reviewers, who made many constructive remarks and suggestions, which we gladly incorporated in this new version of the paper.

References

Blanchon, h. (2004). Comment définir, mesurer et améliorer la qualité, l'utilisabilité et l'utilité des systèmes de TAO de l'écrit et de l'oral. Une bataille contre le bruit, l'ambiguïté, et le manque de contexte. Grenoble, Université Joseph Fourier. Thèse de HDR: 380 p.

Ciravegna, F. (2001). Adaptive information extraction from text by rule induction and generalisation. IJCAI, Seattle.

Daoud, D. M. (2006). It is necessary and possible to build (multilingual) NL-based restricted e-commerce systems with mixed sublanguage and content-oriented methods. Ph.D thesis GETA - CLIPS. Grenoble, Université Joseph Fourier: 296 p.

Hajlaoui, N. (2006). Recherche et production de corpus de messages pour la multilinguisation de sites de e-commerce en SMS, initialement en arabe. proc 6th international IBIMA Conference, Bonn, Allemagne 11 p.

Hajlaoui, N. and C. Boitet (2007). Portage linguistique d'applications de gestion de contenu. TOTh Conférence sur la Terminologie & Ontologie: Théories et Applications, Annecy France 13p.

Hajlaoui, N. (2007). Multilinguïsation de services de gestion de contenu. IC "Ingénierie des Connaissances", plate-forme AFIA "Association Française pour l'Intelligence Artificielle", Grenoble, France 2p.

Huberman, B., P. Pirolli, et al. (1998). Strong Regularities in World Wide Web Surfing. Science vol 280, pp 95-97.

Kittredge R. (1986) Analyzing Language in Restricted Domains. In "Sublanguage Description and Processing", R. Grishman & R. Kittredge, ed., Lawrence Erlbaum, Hillsdale, New-Jersey, 248 p.

Lehman, A. (1996). Construction d'un système de résumé automatique de textes de type scientifique et technique. RECITAL , Paris pp 65-69.

Uchida, H. and M. Zhu (1999). Enconverter Specifications, UNU/IAS UNL Center, 33 p.

Automatic Pronunciation Dictionary Toolkit for Arabic

Hussein Hiyassat[1], Mustafa Yaseen[2], Nihad Arabiat[3]

[1]e-Prucurment Project, UNDP[2]Amman University, [3]Ministry of Education (Amman, Jordan)
E-mail: hussein.hiyassat@undp.org, myaseen@ammanu.edu.jo, nihadahmad@yahoo.com

Abstract

Speech Recognizers are commercially available from different vendors. Along with this increased availability comes the demand for recognizers in many different languages that often were not focused on the speech recognition research. So far, Arabic language is one of those languages. With the increasing role of computers in our lives, there is a demand to communicate with them naturally. Speech processing by computer provides one vehicle for natural communication between man and machine. In this paper, a novel approach for implementing Arabic isolated speech recognition is described. The first SPHINX-IV-based Arabic recognizer is introduced and an automatic toolkit is proposed, which is capable of producing pronunciation dictionary (PD) for both Holly Qura'an and Arabic digits corpus ADC. ADC corpora were tested and evaluated accuracy of 99.213% and WER is 0.787% is obtained.

1 Introduction

With an estimated number of 289 million of Arabic speakers, and since it is the Language of Quran, it is considered as the sacred language of nearly 1.48 billion Muslims throughout the world. There has been little research in the area of Arabic speech recognition compared to other languages of similar or less importance. Due to the lack of diacritic Arabic text and the lack of Pronunciation Dictionary (PD), most of previous work on Arabic Automatic Speech Recognition has been concentrated on developing recognizers using Romanized characters, i.e. the system recognizes the Arabic word as an English one, then maps it to Arabic word from a lookup table. This is of course a shortcoming, and has many deficiencies affecting the recognition and its accuracy. Previous work reported in the literature on Arabic ASR J. Billa et al (2002), J. Billa et al (2002), Katrin Kirchhoff et al (2002),G. Zavagliakos et al (1998), Katrin Kirchhoff et al (2002) aimed at developing recognizers, for either Modern Standard Arabic (MSA) or Egyptian Colloquial Arabic (ECA). Some results of Word Error Rates (WER) obtained from both MSA and ECA are shown below:

Arabic language Type	Year	(WER)
MSA	1997	15-20 %
ECA	96/97	61 -56 %
ECA	2002	55.1-54.9 %

Table 1: Word Error Rates (WER) obtained from both MSA and ECA.

The word error rate of some recognizers under special conditions often is below 10% David S. Pallet, et al (1998), and for general purposes Large Vocabulary Continuous Speech Recognizers (LVCSR) the best word error rates were as high as 23.9% Antti-Veikko Ilmari Rosti(2004), Antti-Veikko Ilmari Rosti(2004) reported for the English language.

Katrin Kirchhoff et al (2002) in their project at the 2002 Johns Hopkins Summer Workshop, focused on the recognition of dialectal Arabic. Three problems were addressed:

1. The lack of short vowels and other pronunciation information in Arabic texts.
2. The morphological complexity of Arabic.
3. The discrepancies between dialectal and formal Arabic.

They used the only standardized corpus of dialectal Arabic available at that time, the LDC Call Home (CH) corpus of ECA. The corpus is accompanied by transcriptions in two formats: standard Arabic script without diacritics and a “Romanized” version, which is close to a phonemic transcription. They stated that Romanized Arabic is unnatural and difficult to read for native speakers; moreover, script-based recognizers (where acoustic models are trained on graphemes rather than phonemes) have performed well on Arabic ASR tasks in the past.

2 Sphinx Engine

SPHINX engine has four versions, namely II,III, IV and pocket SPHINX-IV all of these versions implement the general architecture of speech recognition shown in Figure 1.

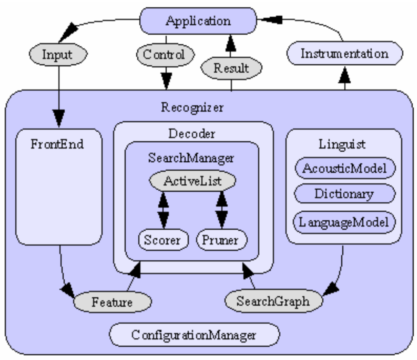


Figure 1: SPHINX-IV architecture

SPHINX-IV engine is customized in this research to handle Arabic language. SPHINX-IV is an open source speech recognition engine built for research purposes by speech research group at Carnegie Mellon University in 1988 by Xuedong Huang et al (2003), Stuart Russell et al (1995), Heidi Christensen (1996), Rabiner, L.R. et al (1993). It was one of the first systems to demonstrate the feasibility of accurate, speaker-independent, large-vocabulary continuous speech recognition. SPHINX is a large-vocabulary, speaker-independent, Hidden Markov Model (HMM)-based continuous speech recognition system. Many Ph.D. dissertations around the world used SPHINX for Speech Recognition of various languages: Jon P. Nedel (2004), Sam-Joo Doh (2000), Yoshiaki Ohshima (1993), Bhiksha Raj (2000), Juan M. Huerta (2000), WilLiam A. Rozzi(1991), FuHua Liu(1994), Michael L. Seltzer(2000), Matthew A. Siegler (1999), none of them addressed Arabic.

3 Pronunciation Dictionaries

One of the core components of a speech recognition system is the Pronunciation Dictionary (PD). Its main purpose is to map the orthographic representation of a word to its pronunciation. The search space of the recognizer is the (PD). The PD lists the most likely pronunciation or citation form of all words that are contained in the speech corpus. The pronunciation of the corpus can range from very simple and achievable with automatic procedures to very complex and time consuming.

3.1 Automatic versus Hand Crafted PD

In order to reduce, both the cost and time required to develop LVCSR systems, the problem of creating (PD) must be solved. In the following sections the process of developing automatic tool for (PD) for Standard Arabic language will be described.

3.2 Orthographic to Phonetic Transcription

Conversion of Arabic phonetic script into rules is one of the major obstacles facing researchers on Arabic text to- speech and speech recognition systems. Although Arabic is one of the oldest languages that its sounds and phonological rules were extensively studied and documented in the old traditional Arabic literature of more than 12 centuries ago, these valuable studies need to be compiled from scattered literatures and formulated in a modern mathematical framework. The aim of this section is to formulate the relation between grapheme to phoneme relationship for Arabic.

4 Grapheme-to-phoneme conversion

To address the conversion of Arabic phonetic script into rules, we will formulate the relationships between grapheme to phoneme for Arabic.

Arabic is an algorithmic language, at least from the phonology, writing and inflection points of view, for example no rules can explain the different pronunciation of "*g*" in English in the following words "*laugh*, *through*, *good* and *geography*", while Arabic language has direct grapheme to phoneme mapping for most grapheme. In general, Arabic text with diacritics is pronounced as it is written using certain rules. Contrary to English, Arabic does not have words with different orthographic forms and the same pronunciation.

There are sixteen basic rules in orthographic to phonetic transcription in Arabic language; Alghamdi, Mansour et al. (2004), those rules are:

1. The sokon sign (ْ), is not a symbol of any phoneme, but it's meaning is this consonant is followed by another consonant, without intermediate vowel, (i.e. if it exists or not it will not affect the pronunciation of the consonant itself). Example "عِلْم" this means that "ل" will be pronounced as is without introducing any vowels.
2. The "ا" after group waw "و" as in "ذهبوا" is not pronounced.
3. Pharyngealization (emphasis): There are Pharyngealized consonants in standard Arabic where the consonant is stressed when it is pronounced.

   Example is the word (عَدَّ) "count" the sign" ّ " here, used as stress when the "د" is pronounced "ع َ د د َ".
4. The pronunciation of *alef* is totally dependent on the successors characters as follows:
   a. Not pronounced if followed by two consonants as in "بالمدرسة " (in the school), this pronounced as " ب ِ ل م َ د ر َ س َ ت ِ".
   b. Pronounced if it is part of the *laam* of definite article as in "القَلَم" the "ا" will be pronounced as follows "ء َ ل ق َ ل َ م".
   c. Pronounced as the short vowel " ُ", if it is the first of a verb, with the third character of it has the short vowel "ُ" as in "اركُض ", in this case it is pronounced as "ء ُ ر ك ُ ض ".
   d. If the above rules did not apply, then the *alef* pronounced as the short vowel " ِ ", as in "ابن" is pronounced as " ء ِ ب ن".
5. The *alef almaqsorah* "ى" , its predecessor is always the short vowel " َ" as in " رمَى" , it is pronounced as "ر َ م َ َ " .

6. Feminine *Taa* ("ة " تاء التأنيث) as in إمرأة which is used in Arabic at the end of a noun to modify its gender from masculine to feminine if the word containing feminine *Taa* found as the last word in sentence then the *Taa* is pronounced as *Haa* " ه" otherwise it is pronounced as "t ت".
7. The letter *laam* (ل) in (ال) is the *Laam* of the definite article, prefixed to nouns; they are added to the structure of the word. There are two types of *Laam* The *Lunar-lam Alqmar* (القمَر) pronounced "ا ل ق م َ ر" a l q a m r" and the *Solar-Lam Alshams* (الشمس) pronounced "Ashmes ا ش ش م س" the *Laam* is not pronounced her. The letter *Laam* in (ال) either pronounced or assimilated depending on the successor character as shown in Table 2.

Pronunciation rule	Successor letter
Lunar-lam (ال) pronounced if it is followed by any of these letters	ء ، ب ، غ ، ح ، ج ، ك ، و، خ ، ف ، ع ، ق ، ي ، م ، ه
Solar-Lam (ال) assimilated if it is followed by any of these letters	ط ، ث ، ص ، ر ، ت ، ض ، ذ ، ن ، د ، س ، ظ ، ز، ش ، ل

Table 2: Pronunciation rules for *laam* (ل).

8. The "*Hamz*" –Glottal-((ء) الهمز): The "*Hamz*" is pronounced when it comes after a pause or at the beginning of an utterance, but it is not pronounced in all other cases.
9. The rules of pronunciation two successive words when the last character of the first and the first character of the second word are not vowels. The general rule is that the short vowel /i/ should be introduced after the last character of the first word.
10. The pause: an utterance in Arabic is never terminated by a short vowel this means that the short vowel of the last word of the sentence is not pronounced.
11. The rule for pronouncing the three "*Tanween*" double diacritic signs (التنوين) namely (ٌ , ً , َ) these diacritics are pronounced as "N" (ن) as in " صبراً " it is pronounced as " ص َ ب رَ ن" if it is not finally or pronounced as " ص َ ب رَ ا" otherwise.
12. The lengthening *alef* " آ ", as in "آية" , is pronounced as " ء َ َ ي َ ه".
13. If the predecessor of the vowel *waw* "و" , is the short vowel " ُ" as in "سُوق ", then it is pronounced as " س ُ ُ ق".
14. If the predecessor of vowel *Yaa* "ي" is the short vowel "ِ" as in " عِيد ", then the *Yaa* is pronounced as " ع ِ ِ د ".
15. If there is three successive consonants, as in مَنْ " "الْقادم, then short vowel " ِ" is introduced as " م َ ن ِ ال ق ا د م ".
16. The *laam* (ل) always in pronounced as a light *laam* called (*traqeeq* ترقيق) except in the pronunciation of the name of Allah (almighty) "الله" or "اللهم" it is then emphasized (*tafkheem* تفخيم) if it comes at the beginning of the utterance or its predecessors either one of the two short vowels " َ ", " ُ " .

5 Arabic Digits Corpus (ADC)

the ADC is built for developing Arabic digit model recognition for digits zero, one . . . nine " صفر, واحد . . . تسعة " . The ADC corpus is developed using the recording of 142 speakers, table 3 shows the details of the speakers. The ADC Consists of two disjoint sets of utterances: 1213 training utterances collected from 73 male and 49 female speakers, and 143 testing utterances from 12 male and 8 female speakers .The total length of the training utterances is about 0.67 hr.

Gender	Training	Testing	Total
Male	73	12	85
Female	49	8	57

Table 3: speakers participated in ADC corpus.

Baseline system is trained using about 35 minutes of speech, including all conditions together. About 7 minutes of evaluation data are used for the test in this paper.

6 Senones Clustering

Triphones modeling assumes that every triphone context is different. phonetic and sub phonetic units share parameters at unit level, each cluster represents a set of similar Markov states and is called a Senone. A sub word model composed of a sequence of senones after the clustering is finished. The optimal number of senones for a system is mainly determined by the available training corpus and can be tuned on a development set. In our experiments 500, 1000, 3000 and 5000 senones are examined. Results are shown in table 4.

No. of Senones	WER %	Accuracy %
500	7.2	90.4
1000	0.8	99.2
3000	4.2	94.1
5000	4.0	93.8

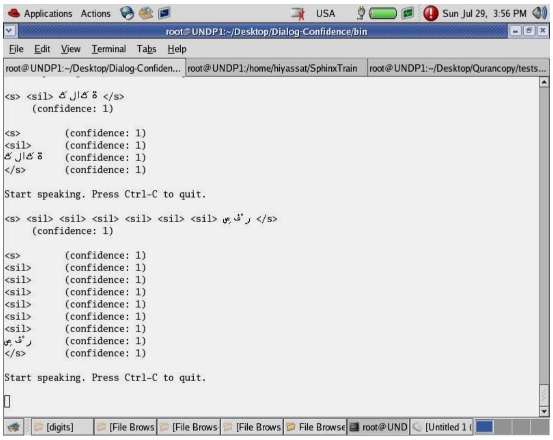
Table 4: Number of senones versus WER for ADC.

Table 4 shows that the best WER is achieved at 1000 senones for the ADC. The over all performance tests results is shown in figure 2 taken from output of the system.

```
Performance test of the digits corpus
Accuracy: 99.213%   Errors: 1  (Sub: 0
Ins: 0  Del: 1)
Words: 127  Matches: 126   WER: 0.787%
Sentences:  127        Matches:  126
SentenceAcc: 99.213%
This  Time Audio: 1.39s  Proc: 0.09s
Speed: 0.07 X real time
Total Time Audio: 143.24s  Proc: 9.91s
Speed: 0.07 X real time
Mem  Total: 126.62 Mb  Free: 114.17 Mb
Used: This: 12.46 Mb Avg: 12.76 Mb Max:
18.44 Mb
===================================
```

Figure 2 Test summaries for the ADC

Screen shot of the digit recognition of the ADC is shown in figure 3.


Applications Actions
USA
Sun Jul 29, 3:56 PM
root@UNDP1:~/Desktop/Dialog-Confidence/bin
File Edit View Terminal Tabs Help
root@UNDP1:~/Desktop/Dialog-Confiden...
root@UNDP1:/home/hiyassat/SphinxTrain
root@UNDP1:~/Desktop/Qurancopy/tests...
[digits]
[File Brows
[File Brows
[File Brows
File Browse
root@UND
[Untitled 1 (

```
<s> <sil> ث ال ث ة </s>
     (confidence: 1)

<s>         (confidence: 1)
<sil>       (confidence: 1)
ث ال ث ة    (confidence: 1)
</s>        (confidence: 1)

Start speaking. Press Ctrl-C to quit.

<s> <sil> <sil> <sil> <sil> <sil> <sil> ص ف ر </s>
     (confidence: 1)

<s>         (confidence: 1)
<sil>       (confidence: 1)
<sil>       (confidence: 1)
<sil>       (confidence: 1)
<sil>       (confidence: 1)
<sil>       (confidence: 1)
<sil>       (confidence: 1)
ص ف ر       (confidence: 1)
</s>        (confidence: 1)

Start speaking. Press Ctrl-C to quit.
```

Figure 3: output of the performance test for ADC

7 Conclusion

SPHINX-IV has been used as engine for Arabic speech recognition, building speech recognition resources for Arabic, and building Automatic Pronunciation Dictionary Toolkit (APDT), this tool didn't exist before in SPHINX-IV or any other speech recognition engine. Three corpuses were built using this tool. It is found that this tool is suitable for Arabic Recognition. Accuracy obtained for ADC is 99.213% and WER is 0.787% which are much better than the results obtained by M. M. El Choubassi et al (2003), where results were reported to achieve 87% accuracy. Results are very satisfactory taken into account the size of the corpus of training which was used if compared with corpora used for English.

8 References

Alghamdi, Mansour et al. (2004). Arabic Phonological Rules, Journal of King Saud University: Computer Sciences and Information. 16: 1-25. (in Arabic).

Antti-Veikko Ilmari Rosti(2004). Automatic transcription of conversational telephone speech - development of the CU-HTK 2002 system. Technical Report CUED/F-INFENG/TR.465, Cambridge University Engineering Department, 2003, Available at http://mi.eng.cam.ac.uk/reports/.

Antti-Veikko Ilmari Rosti(2004). Linear Gaussian Models for Speech Recognition, Ph.D. Thesis, Wolfson College, University of Cambridge, May 2004.

Bhiksha Raj (2000). "Reconstruction of Incomplete Spectrograms for Robust Speech Recognition", Ph.D. Thesis, Department of Electrical and Computer Engineering , CARNEGIE MELLON UNIVERSITY, April 2000.

CMU Sphinx Open Source Speech Recognition Engines, URL: http://www.speech.cs.cmu.edu/

David S. Pallet, et al. (1998) Broadcast News Benchmark Test Results". In Proceedings of the DARPA Broadcast News Workshop, Herndon, Virginia, February 28-March 3, 1999

Fu-Hua Liu(1994). “Environmental Adaptation for Robust Speech Recognition", PhD Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 1994.

G. Zavagliakos et al.(1998). “The BNN Byblos 1997 large vocabulary conversational speech recognition system,” in Proceedings of ICASSP, 1998.

Heidi Christensen(1996). Speaker Adaptation of Hidden Markov Models using Maximum Likelihood Linear Regression, Ph.D. Thesis, Institute of Electronic Systems Department of Communication Technology ,Aalborg University.

J. Billa et al.(2002). “Arabic speech and text in Tides

On Tap," in Proceedings of HLT, 2002.

J. Billa et al.(2002). "Audio indexing of broadcast news," in Proceedings of ICASSP, 2002.

Jon P. Nedel,(2004). Duration Normalization for Robust Recognition of Spontaneous Speech via Missing Feature Methods, Ph.D. Thesis, Department of Electrical and Computer Engineering , CARNEGIE MELLON UNIVERSITY, April, 2004.

Juan M. Huerta (2000). "Robust Speech Recognition in GSM Codec Environments", Ph.D. Thesis, Department of Electrical and Computer Engineering, CARNEGIE MELLON UNIVERSITY, April 2000.

Katrin Kirchhoff et al. (2002). "Novel Approaches To Arabic Speech Recognition", the 2002 johns-hopkins summer workshop,2002.

M. M. El Choubassi et al. (2003). "Arabic Speech Recognition Using Recurrent Neural Networks" ,isspit 2003

Matthew A.Siegler (1999). "Integration of Continuous Speech Recognition and Information Retrieval for Mutually Optimal Performance", Ph.D. Thesis, Department of Electrical and Computer Engineering, CARNEGIE MELLON UNIVERSITY, 1999.

Michael L. Seltzer(2000). "Automatic Detection Of Corrupt Spectrographic Features For Robust Speech Recognition ", Master degree Thesis, Department of Electrical and Computer Engineering, CARNEGIE MELLON UNIVERSITY, May 2000.

Rabiner, L.R. et al.(1993). "Fundamentals of Speech Recognition", Prentice-Hall, New Jersey.

Sam-Joo Doh (2000). Enhancements to Transformation-Based Speaker Adaptation: Principal Component and Inter-Class Maximum Likelihood Linear Regression, Ph.D. Thesis, Department of Electrical and Computer Engineering , CARNEGIE MELLON UNIVERSITY, July 2000.

Stuart Russell et al.(1995). "Local Learning In Probabilistic Networks With Hidden Variables", (1995), Computer Science, IJCAI.

Xuedong Huang et al.(2003). "The SPHINX-II Speech Recognition System: An Overview ", School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213, 2003.

WilLiam A. Rozzi(1991). "Speaker Adaptation in Continuous Speech Recognition via Estilsiation of Correlated Mean Vectors", PhD Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 1991.

Yoshiaki Ohshima (1993). Environmental Robustness in Speech Recognition using Physiologically-Motivated Signal Processing, Ph.D. Thesis, Department of Electrical and Computer Engineering, CARNEGIE MELLON UNIVERSITY, December 1993.

Broadcast News Transcription Baseline System using the NEMLAR database

R. Bayeh[1 2], C. Mokbel[2], G. Chollet[1]
[1] TELECOM-ParisTech, CNRS-LTCI UMR-5141
46 rue Barrault, 75634 Paris cedex 13, France

[2] University of Balamand
PO Box 100, Tripoli, Lebanon

E-mail: {bayeh,gerard.chollet}@TELECOM-ParisTech.fr,{rbayeh,cmokbel}@balamand.edu.lb

Abstract

This paper describes one of the first uses of the NEMLAR Arabic Broadcast News Speech Corpus (BNSC) for the creation of an automatic speech recognizer (ASR) for Arabic Broadcast News (BN). Different parameterization settings, types of acoustic models, various language models and testing schemes are presented for the creation of a baseline system for Modern Standard Arabic using the NEMLAR BNSC database. To port this system to dialects, a certain amount of dialectal data is required. Due to the absence of such resources and the use of other languages in dialectal speech, techniques for the creation of cross-lingual models using the baseline system are investigated. Certain techniques that have given promising results in previous experiments are proposed. These techniques, which would be helpful in developing a cross-dialectal speech recognition system, have been, due to the use of Maghrebian/Levantine dialects which make use of French, experimented in a cross-lingual Arabic-French frame. Although a lot of work remains to be accomplished, the current results are very encouraging.

1. Introduction

Arabic, which exists in many forms or dialects, is spoken by almost 250 million people, thus making it one of the most widely spoken languages in the world. The main classes of Arabic dialects are the Eastern and the Western dialects. The Eastern dialects are spoken in Egypt, Sudan and the Middle East while the Western are spoken in North African regions such as Tunisia, Morocco, Algeria, etc… Both of these classes can be further divided into even smaller classes such as Levantine, Maghrebian, Gulf… However, even within the same class, a difference can be perceived among the dialects and for this reason, Arabic speakers often rely on Modern Standard Arabic (MSA) to communicate when their native dialect is not understood by their listeners. MSA, considered the universal form of Arabic, is not spoken as a mother tongue but exists in written un-vowelized (no representation of short vowels) form. Moreover, MSA is the only written form of Arabic since none exists for dialects. Despite this and MSA's ample use in broadcast news, literature, newspapers, and electronic texts, the efforts made for the creation of electronic Arabic corpora with proper transcriptions and phonetic representations were fewer than those made for the creation of other corpora for other languages (Schultz, 2006).

One important "Arabic" effort, recently produced by the NEMLAR partners, is the Arabic broadcast news speech corpus comprising of 40 hours of MSA (Yaseen, 2006). This corpus also includes segments of dialectal speech, some of which were included in our experiments. The following section describes both databases and the data preparation procedure undertaken to determine which segments are to be used for the development of the Arabic BN transcription systems. Section 3 provides the details of the baseline system (parameterization, acoustic modeling, language resources…). A cross lingual approach is also reported. Then, the results obtained so far are presented and discussed in section 4. Finally, some conclusions and perspectives are given.

2. Database and Data Preparation

The NEMLAR Arabic Broadcast News Speech Corpus consists of approximately 46 hours of data, 40 hours of which are transcribed in MSA and the rest include jingles, music segments, speech/music overlaps, un-transcribed segments and some dialectal spontaneous speech etc… This data is the collection of four different radio stations' daily broadcasts, 25 to 30 minutes each. The topics covered in these broadcasts include general news, political news, sports events, interviews, political debates. The speakers, on the other hand, are of various nationalities mainly Lebanese, Moroccan, Egyptian. Therefore the underlying dialect behind the MSA is Levantine/Maghrebian. A more detailed description of the database can be found in (Yaseen, 2006).

Although the dialectal form is rarely used in the database, its existence cannot be ignored. Dialects affect the pronunciation of certain words even if MSA is being spoken. For example, the "j" in any word is pronounced differently by an Egyptian speaker, a Gulf speaker and a Lebanese speaker. This is an issue in the database, because it is not indicated in the transcription.

The transcriptions were first analyzed to remove the un-transcribed segments, the music/jingle/noise segments as well as the overlapping segments. As for the dialectal segments, these were selected based on the clarity of speech/transcription depending on human expertise. The final number of hours used is 42 hours of clearly transcribed MSA/dialectal speech, approximately 40 of which were used for training. These dialectal segments were included to increase the robustness of acoustic models if they are to be used for cross-lingual experiments later as proposed in the next section.

3. Baseline System for BN ASR

4.1 Corpus Definition

The NEMLAR database was manually transcribed using the Transcriber software and validated. The first step after its validation was selecting the useful segments mentioned above (non-jingle, non-music…) and creating the necessary resources such as the language model and the pronunciation dictionary. Few inconsistencies of transcriptions for the same word were found in the database. We cite especially the case of the vowelized "Al" prefix which has been transcribed in different ways to represent the same acoustical content (Aol, Aal, AAal,…) Such inconsistencies were reduced as best as possible for better modeling results.

The morphology is rich in the Arabic language. Taking into account some of morphological characteristics while defining the vocabulary would certainly improve the results. Arabic words can usually be decomposed into a root word with a prefix and suffix. This decomposition follows some complex rules that, if included in the dictionary creation process, improve results (Messaoudi, 2005). However, for the time being, dictionaries were created directly from the database's provided transcriptions. These transcriptions take into consideration short vowels thus rendering the creation of the pronunciation dictionary very easy. The words were simply expanded as shown in the example below:

Table 1: Vowelized representation and expansion

Word Representation	Word Expansion	Definition
Kataba كَتَبَ	k a t a b a	*he wrote*
Kutubo كُتُبْ	k u t u b o	*books*

The dictionary created, made up of 59K completely vowelized words (i.e. a word representation includes short vowels) and was used for training. Two other dictionaries were then created from this vowelized version and used for testing. The first of these versions was the unvowelized version (~39k unique words, average of 1.5 expansions/word). This was created by simply removing the short vowels from the word representation as shown in Table 2.

Table 2: Un-vowelized representation and vowelized expansion

Word Representation	Word Expansion
ktb كتب	k a t a b a
ktb كتب	k u t u b o

Then, an automatically diacriticized version of the unvowelized dictionary was created by appending the missing word expansions with different short vowels at the end for each word (approx 1k words+expansions were added). For the example above the following words are added in Table 3. In this version, the words added are not always grammatically correct however this was done because error analysis of previous results indicated confusion of short vowel endings. A more precise version of the dictionary is to be created and tested with the help of a phonetician at a later stage.

Table 3: Un-vowelized representation and automatically vowelized expansions

Word Representation	Word Expansion
ktb كتب	k a t a b a
ktb كتب	k a t a b i
ktb كتب	k a t a b u
ktb كتب	k a t a b o
ktb كتب	k u t u b a
ktb كتب	k u t u b i
ktb كتب	k u t u b u
ktb كتب	k u t u b o

4.2 Language Resources

Almost all written resources do not include short vowels in the word representations. Therefore, all the transcriptions were un-vowelized as was done to create the dictionary and several bigram language models were created using the toolkit SRILM (Stolcke, 2002). For our first tests, two language models were created from the NEMLAR transcriptions; a vowelized and an un-vowelized language model. The size of the NEMLAR text is around 250 K words, which is too limited to create a precise language model. Therefore, a language model based on the NEMLAR transcriptions appended with the An-Nahar database, which comprises of approx 1.3M words (~26k unique words), was created and made to include the NEMLAR vocabulary.

4.3 Parameterization and Acoustic Modeling

Filter bank parameterization was conducted on all the selected portions of data to create Mel-Frequency Cepstral Coefficient (MFCC) vectors with 12 cepstrum coefficients, log energy and the first and second order derivatives. The cepstra were computed on windows of 20 ms duration every 10 ms with no band limiting obtained using the Hamming windowing technique. Additional preprocessing is also applied at this point, i.e. cepstral mean (CMS) and variance normalization in order to take into account the channel effects.

As a consequence of the two parameterization schemes implemented, several sets of 3-state right to left Hidden Markov (HMM) acoustic models were developed. Each set includes models for 37 Arabic phones; 28 consonants and 6 vowels (3 short, 3 long), and 3 non-linguistic units (silence, short pause, and miscellaneous [inhalation, exhalation…]). A total of three sets of acoustic models, shown in Table 4 below, have been created.

Table 4: 3 state HMM acoustic models

Nb. Gaussians	Parameterization	Context (in)dependent
256	Not including CMS and variance normalization	independent
256	Including CMS and variance normalization (not all results are out yet)	independent
32	Not including CMS and variance normalization	Dependent (triphone)
32	Including CMS and variance normalization	Dependent (triphone)

An embedded training version of the Baum-Welch algorithm in the HTK Toolkit was used for the creation of all these models.

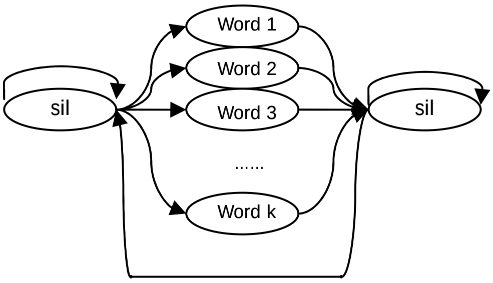
4.3 Cross-Lingual Modeling

After a baseline was developed for Modern Standard Arabic, cross-dialectal acoustic models were to be taken into considerations. Due to the lack of resources necessary for the development of such models, cross-lingual French-Arabic acoustic models were investigated. The utility of such models in voice commands application is shown in (Bayeh, 2004). It should be noted here that it was taken into consideration that the Maghrebian and Levantine dialects include diglossia and code-switching using French words. For the creation of a cross lingual model, the phones of both languages should be compared. This results in the division of each language's phones into language specific and shared sets. At this point, a common set of acoustic units that can be used for cross lingual recognition can be formed by joining the language specific phones of each language, and adapting new models for the shared phones. The models for the shared phones are a combination of the models from the two languages that can be reduced further by tree classification algorithm (Bayeh, 2004).

4. Results and Discussions

All models developed were tested using recursive word loops with optional start and end silence nodes as shown in Figure 1.

Figure 1: Recursive loop of K words



The results obtained from our experiments are expressed in terms of correct rate, which is calculated as follows:

$$\%Correct = \frac{N - D - S - I}{N} \times 100$$

Where *N* is the total number of transcriptions, *D* is the number of deletion errors, *S* is the number of substitution errors, and *I* is the number is of insertion errors (Odell, 2002).

Initially, some preliminary experiments were conducted using a 5k words recursive loop to test the parameterization settings. It was very evident that those including cepstral mean and variance normalization yielded better results as shown in Table 5.

Table 5: Comparison of the parameterization settings

Model	Network	Language Model	Correct Rate
32 gaussians context-dependent triphone model without normalization	5 K wds vowelized	No	73.48 %
32 gaussians context-dependent triphone model with normalization	5 K wds vowelized	No	76.71 %

However, the French models were created using parameterization settings without normalization. Therefore, in order to complete our cross-lingual experiments, the models resulting from the parameterization without normalization were used in the rest of the experiments in order to determine the best set of models. The number of words in the test loop was incremented to 20k and several tests were conducted. The results are shown in Table 6.

Table 6: Results using a vowelized dictionary for training and testing (parameterization without normalization)

Model	Network	Language Model	Correct Rate
256 gaussians context-independent monophone model	5 K wds vowelized	No	53.24 %
256 gaussians context-independent monophone model	20 K wds vowelized	No	37.40 %
256 gaussians context-independent monophone model	20 K wds vowelized	20K LM	41.20 %
256 gaussians context-independent monophone model	59 K wds vowelized	No	28.49 %
32 gaussians context-dependent monophone model	20 K wds vowelized	No	62.55 %
32 gaussians context-dependent monophone model	20 K wds vowelized	20K LM	64.37 %

In addition, the vowelization effect during testing was studied. When the test loop consists of un-vowelized words, the performance of the system decreases. This is attributed to the influence of the speakers' different dialects on the pronunciation of short vowels and thus the confusion of some of these sounds.

Table 7: Comparison of vowelized / unvowelized / automatically vowelization techniques

Model	Network	Language Model	Correct Rate
32 gaussians context-dependent monophone model	20 K wds vowelized	20K LM	64.37 %
32 gaussians context-dependent monophone model	20 K wds Un-vowelized	20K LM	62.79 %
32 gaussians context-dependent monophone model	20 K wds Aut.vowelization	20K LM	67.44 %
32 gaussians context-dependent monophone model	20 K wds Aut.vowelization	Full LM (NEM + Nahar)	78.57 %

As an attempt to deal with this, a cross-lingual model was created using the obtained baseline system. The Arabic vowel models were copied and adapted using French models as described in (Bayeh, 2004) for the creation of this new model set and 20k vowelized test loops were used to test it. The results yielded are shown in Table 8. These results show that the selection of phones for this copying technique can increase the performance of a system.

Table 8: Context-dependent models initialized using French models

Model	Network	Language Model	Correct Rate
NEMLAR 32 gaussians context-dependent triphone model without normalization	20k wds vowelized	20K LM	64.37 %
NEMLAR/French 32 gaussians context-dependent triphone model with normalization	20 K wds vowelized	20K LM	75.41 %

The results obtained for the systems trained using the NEMLAR database, as presented in this paper, are very comparable to the state of the art results shown in (Messaoudi, 2004). In addition, when evaluating the correct rate, the recognition transcriptions obtained are compared to those initially provided by NEMLAR. We notice, during this comparison, that a substantial number of words beginning with the “Al” prefix are confused with the similar word but without the prefix. This is due to the fact that sometimes the “Al” is not pronounced as Al but just as A (this can probably be solved if grammar rules are taken into consideration). Another issue is the insertion of prefixes such as “waw”, “bi”, “li”. An error that is specific to the vowelized tests is the one concerning the “hamza” on top of the “Alef”. Unlike the “Al” case, the “hamza” is often pronounced but not transcribed. If these issues are taken into consideration when evaluating the results, the correct rate can be improved even further.

5. Conclusion and Perspectives

In this paper, a novel database, the NEMLAR Broadcast News Speech Corpus, was used for the development of an ASR system. Results of approximately 80% correct rate were yielded. In addition, this paper proposes a unique approach to deal with dialectal speech in Broadcast News speech recognition. As mentioned earlier, the use of MSA is preferred in ASR because of the few if nonexistent resources for some dialectal forms of the language. In addition, Arabic dialects can render the development of robust acoustic models from scratch difficult due to the existence of diglossia and code-switching. These are interesting aspects and greatly support the need for other types of acoustic models such as the ones proposed in Section 3. The results obtained show that such models could later aid in the development of automatic transcription systems for regional dialects where only small databases exist to train precise systems. As shown, this can be done as proposed in (Bayeh, 2004) by porting from a highly resourced language to another less resourced language by selecting certain phones.

6. References

Bayeh, R., Lin, S., Chollet G., Mokbel C. (2004). Towards Multilingual Speech Recognition Using Data Driven Source/Target Acoustical Units Association, *In the Proceeding of the International Conference of Acoustic Signals and Systems Processing (ICASSP)* Vol I, pp. 521-524.

Messaoudi, A., Lamel, L., Gauvain, J.L. (2005). Modeling vowels for Arabic BN transcription, *INTERSPEECH-2005,* pg. 1633-1636.

Messaoudi, A., Lamel, L., Gauvain, J.L. (2004). Transcription of Arabic Broadcast News, *In the Proceedings of the International Conference on Spoken Language Processing (ICSLP),*

Odell, J., Ollason, D. Woodland, P., Young, S., Jansen, J. (2002). *The HTK Book for HTK V3.2,* Cambridge University Press, Cambridge, UK.

Schultz T., Kirchhoff K. (2006). *Multilingual Speech Processing,* Elsevier, USA, 2006

Stolcke, A., SRILM -- an extensible language modeling toolkit, *In the Proceedings of the International Conference on Spoken Language Processing (ICSLP),* pg. 901-904.

Yaseen M. et al. (2006). Building Annotated Written and Spoken Arabic LR 's in NEMLAR Project, *In the Proceedings of the Language Resource and Evaluation Conference (LREC),* pg. 533-538.

Arabic-English translation improvement by target-side neural network language modeling

Maxim Khalilov, José A. R. Fonollosa
F. Zamora-Martínez, María J. Castro-Bleda, S. España-Boquera

Centre de Recerca TALP, Universitat Politècnica de Catalunya
Barcelona, Spain

Dep. de Lenguajes y Sistemas Informáticos, Universidad Politécnica de Valencia
Valencia, Spain

Abstract

The quality of translation, produced by Statistical Machine Translation (SMT) systems, crucially depends on generalization, provided by the statistical models involved into translation process. In this study we present the n-gram based translation system (i.e. the UPC SMT system), enhanced with a continuous space language model (LM), estimated with a neural network (NN). In the framework of the study, we use NN LM on the rescoring step, reevaluating the N-best list of complete translations hypothesis. Different word history length included in the model (n-gram order) and distinct continuous space representation (i.e. including words, appearing in the training corpus more than *k* times) are considered in the paper. We report result for an Arabic-English translation task, improving Arabic-English translation accuracy by better target language model representation in contrast with the state-of-the-art approach. The experimental results are evaluated by means of automatic evaluation metrics correlated with fluency and adequacy of the generated translations.

1. Introduction

Language modeling is an essential step in many Natural Language Processing applications, and, particularly in the Statistical Machine Translation (SMT) task.

The most widespread and popular technique for language model estimation in the state-of-the-art SMT systems is the so-called n-gram approach, which assign high probability to frequent sequences of words by considering the history of only n-1 preceding words in the utterance. On the contrast, the approach presented in the paper can be considered as a coherent and natural evolution of the probabilistic Language Model (LM): we propose to use a continuous LM trained in the form of a neural network (NN). The idea of continuous space representation of language is not new, some successful attempts of NN model application to language modeling has been recently made (Xu and Jelinek, 2004; Bengio et al., 2003; Castro and Prat, 2003). However, the use of NN LM in the state-of-the-art SMT systems is not so popular, due to its high computational cost. The only comprehensive work refers to (Schwenk et al., 2006b; Schwenk et al., 2006a), where the target LM is presented in the form of fully-connected multi-layer perceptron.

Our work addresses the improvement of Arabic to English translation which is considered as a complex translation task since high dissimilarity between the source and the target languages. Classical Arabic, like Modern Standard Arabic, is a VSO (verb-subject-object) pro-drop language with rich templatic morphology where words are made up of roots and affixes and clitics agglutinate to words, while English follows the SVO (subject-verb-object) word order and is a non-pro-drop language with less affluent morphology, but with a higher number of irregular verbs.

As mentioned above, NN language models extremely demand for memory resources and computational time. Considering this limitation and be aimed to operate on a fair experimentation field demonstrating a pure impact generated by the target NN LM, we use a 30 K phrases extraction from the NIST 2008[1] corpus, belonging to the news domain (*Newswire*), that can be characterized as extremely limited amount of training data (about 700 K of tokens in the English part and 640 K in the Arabic part).

The basic idea of the study lies in improving of the Arabic-English translation quality, circumventing difficulties imposed by complex structure of the Arabic language. Instead of dealing with Arabic we improve the target language (English) model representation, hopefully improving translation fluency without impairing of adequacy.

The article is structured as follows: in Section 2 we describe the NN LM and give a brief explanation of the training algorithm. In Section 3 we outline the n-gram-based SMT system, Section 4 presents our experimental setup and obtained results, while Section 5 concludes the article with the leading discussions.

2. Neural network language model

Our approach to the widely-used statistical language models based on n-grams consists on using neural networks. A NN LM is a statistical LM which follows the same equation as n-grams:

$$p(w_1 \dots w_{|W|}) \approx \prod_{i=1}^{|W|} p(w_i | w_{i-n+1} \dots w_{i-1}) \qquad (1)$$

and where the probabilities that appear in that expression are estimated with a NN. The model naturally fits under the probabilistic interpretation of the outputs of the NNs: if a NN is trained as a classifier, the outputs associated to each class are estimations of the posterior probabilities of the defined classes. The demonstration of this assertion can be found in a number of places, for example, in Bishop (1995).

[1] http://www.nist.gov/speech/tests/mt/2008/

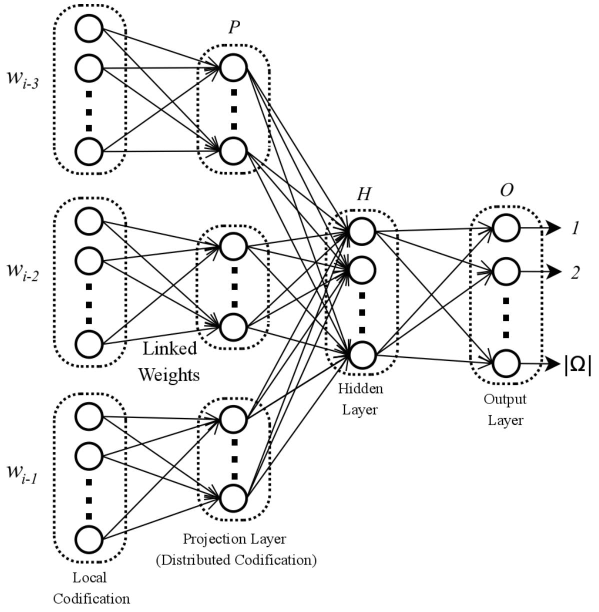


Figure 1: Architecture of the continuous space NN LM. The input words are $w_{i-n+1}, \ldots, w_{i-1}$ and P, H and O are the projection, hidden and output layer, respectively.

The training set for a LM is a sequence $w_1 w_2 \ldots w_{|W|}$ of words from a vocabulary Ω. In order to train a NN to predict the next word given a history of length n-1, each input word must be codified. A natural representation is a local codification following a "1-of-$|\Omega|$" scheme. The problem of this codification for tasks with large vocabularies (as is the case) is the huge size of the resulting NN. We have solved this problem following the ideas of Bengio et al. (2003), learning a distributed representation for each word.

Figure 1 illustrates the architecture of the feed-forward NN used to estimate the NN LM. The input is composed of words $w_{i-n+1}, \ldots, w_{i-1}$ of equation (1). Each word is represented using a local codification. P is the projection layer of the input words, formed by $P_{i-n+1}, \ldots, P_{i-1}$ projection units, which represent the distributed codification of each input word. H denotes the hidden layer and the output layer O has $|\Omega|$ units, one for each word of the vocabulary. This NN, trained as a classifier, predicts the posterior probability of each word of the vocabulary given the history, i.e., $p(w_i|w_{i-n+1} \ldots w_{i-1})$.

In order to achieve a good configuration (topology and parameters) for each NN LM in the translation task, exhaustive scanning using a tuning set was performed. The activation function for the hidden layers was the *hyperbolic tangent* function and the *softmax* function for the output units. Best configurations used a projection layer of 32 units for each word. As an example of the huge sizes of the NNs used, the best experiment (see Table 3) used a 4-gram NN LM and a vocabulary of 4 908 tokens (words with less than k=10 occurrences were discarded). This NN had 157 088 weights, replicated n-1 times at the projection layer, and 325 228 weights at the hidden and output layers.

3. UPC n-gram SMT system

A detailed description of the architecture of the n-gram UPC translation system, which was used in the work, can be found in Crego et al. (2006) and in Mariño et al. (2006). Using *noisy-channel* and *maximum entropy* approaches, it is possible to combine additional feature models in the determination of the translation hypothesis, as shown below:

$$\hat{e}_1^I = \arg\max_{e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \qquad (2)$$

where the feature functions h_m refer to the system models, namely *bilingual translation model*, *target LM* and additional feature models (a *word penalty model*, a *Part-of-Speech (POS) target LM*, a *source-to-target* and a *target-to-source* lexicon models (Och et al., 2004)); the set of λ_m refers to the weights corresponding to these models.

The n-gram translation system is based on a bilingual model, constituting of bilingual units (called tuples). This model approximates the joint probability between the source and the target languages under consideration. The procedure of tuples extraction from a word-to-word alignment (performed with GIZA++ (Och and Ney, 2000)) according to certain constraints is explained in detail in Mariño et al. (2006). In this way the context used in the translation model is bilingual, it not only takes the target sentence into account, but both languages linked in tuples. The translation model can be seen here as a LM, where the language is composed by tuples.

The decoder (called MARIE), an open source tool[2], implementing a beam search strategy based on dynamic programming and allowing for distortion capabilities was used in the translation system. It takes into account all the feature functions described above, along with the bilingual n-gram translation model. It allows for histogram and threshold pruning and hypothesis recombination.

n-gram-based translation system is highly sensitive to the difference in word order between source and targer languages, because of this reason extended monotone distortion model based on automatically extracted reordering patterns was introduced, as presented in Crego and Mariño (2006).

Given the development set and references, the log-linear combination of weights was adjusted using a *simplex* (Nelder and Mead, 1965) optimization method (with the optimization criteria of the highest hybrid metric, based on BLEU (Papineni et al., 2002) and NIST scores (Doddington, 2002)) and an N-best re-ranking just as described in *http://www.statmt.org/jhuws/*. This strategy allows for a faster and more efficient adjustment of model weights by means of a double-loop optimization, which provides significant reduction of the number of translations that should be carried out.

3.1. Arabic data preprocessing

We used a similar approach to that shown in Habash and Sadat (2006), we used the MADA+TOKAN system for disambiguation and tokenization. For disambiguation only

[2] http://gps-tsc.upc.es/veu/soft/soft/marie/

diacritic unigram statistics were employed. For tokenization we used the D3 scheme with -TAGBIES option. The scheme splits the following set of clitics: w+, f+, b+, k+, l+, Al+ and pronominal clitics. The -TAGBIES option produces Bies POS tags on all taggable tokens.

3.2. Rescoring

The NN LM was used on the rescoring/reranking step. Firstly the N-best list of possible translations is generated from the output lattice of the MARIE decoder (N=1 000). On the second step, the N-best translation hypotheses is reevaluated by adding additional features to the baseline (i.e. NN LM) and discriminatively reranking the translation hypothesis according to the log-linear approach. It allows to obtain a better generalization ability. This feature should be able to better distinguish between higher and lower quality translations.

4. Experiments

The experiment results were obtained on the 30 K phrases extraction from the NIST'07 corpus. Automatic evaluation conditions were case-sensitive with tokenized punctuation marks. The development and test sets were provided with four reference translations and contain 489 and 500 sentences, respectively. A brief statistics of the training corpus in use can be found in table 1.

	Arabic	English
Sentences	30 K	30 K
Words	839.59 K	936.39 K
Average sentence length	27.99	31.22
Vocabulary	43.39 K	33.07 K

Table 1: Basic statistics of the training corpus (30 K extraction from the NIST'08 Arabic-English corpus).

4.1. Baseline

MARIE decoder was used to generate a word lattice which is used afterward to extract the 1 000-best list. The optimization criteria was 100*BLEU + 4*NIST, following the point from (Chen et al., 2005). A target language LM was generated using the SRI Language Modeling Toolkit[3] (Stolcke, 2002) on the basis of the considering vocabulary (see Table 1). The following LM configuration was chosen based on the lower perplexity principle (perplexity values were estimated on the concatenation of the development set references): a *4-gram* model with *unmodified Kneser-Ney back-off* discounting and counts post-modification after discount estimation (*-kn-modify-counts-at-end* option). This LM was implicitly integrated into the SMT system as a feature function and was considered as a reference baseline. In case of baseline system we did not perform any additional rescoring, instead of it, we extracted the single-best list corresponding to the highest cost that was considered as a translation.

4.2. NN LM experiments

The NN LM models were trained with the `April` toolkit (España-Boquera et al., 2007), developed for neural networks and pattern recognition tasks in the investigation group of Valencia. Target NN LMs were trained on exactly the same training data as the baseline LM. We considered two key parameters of the continuous NN LM: (a) *word frequency threshold* k: words with less than k occurrences were discarded; (b) n*-gram order*: 3-gram, 4-gram and mixed 3-and 4-gram models were tested.

For the mixed 3-and 4-gram models, several coefficients to combine both models were tested on the tuning set, as shown in equation (3). The best performance was achieved with $\alpha = 0.5$, that corresponds to an equally weighted linear combination of the models:

$$p(w_1 \dots w_{|W|}) \approx \prod_{i=1}^{|W|} \alpha\, p(w_i|w_{i-2}w_{i-1}) + (1-\alpha)\, p(w_i|w_{i-3}w_{i-2}w_{i-1}) \quad (3)$$

1 000-best lists generated from the word lattice were rescored with the NN LM. In order to avoid problems of possibly imperfect optimization, different start points were tried and the best set of weights due to the 100*BLEU + 4*NIST criteria was chosen.

In order to reduce the time needed for the rescoring, we made an attempt to decrease the size of the models, extracting the part related to the distributed codification into an auxiliary table (associating a unique code to each word), and on the other hand maintaining the structure of the net that calculate the LM without codification part. It allowed to reduce the number of the weights by up to half.

4.3. Results

Table 2 shows the perplexity values, obtained on the concatenation of all the references of the development corpus applied to the LMs estimated using SRI LM toolkit and NN techniques. The vocabulary of the k=10 systems is 4 908 words, while the k=12 models include 4 398 words. BLEU, NIST and METEOR scores, as well as the baseline system translation quality comparative results, are reported in Table 3. *Baseline* refers to the SMT systems with the SRI target LM, including all vocabulary words (see subsection 4.1.).

Our previous experience shows that, at least for small translation tasks with a lack of training material, target LM perplexity reduction leads to a notable improvement in translation quality. As can be observed, considerable improvements were obtained by using a reranking of the 1 000-best list with NN LM: for the major part of the considered configurations of NN LMs, BLEU score is higher than for the baseline configuration. Statistical significance threshold[4] lies on the level of 27.40 and 22.05 BLEU points for the development and test sets, respectively. Consequently, incorporating of NN LM technique to the n-gram-based SMT

[3] http://www.speech.sri.com/projects/srilm/

[4] We used a bootstrap resampling method as described in (Koehn, 2004) for the 95% confidence interval and 1 000 set resamples.

	3-gram		4-gram		3- and 4-grams mixture	
k	10	12	10	12	10	12
SRI LM	106.55	103.17	111.49	108.19	–	–
NN LM	97.75	93.14	98.11	92.32	94.89	89.73

Table 2: Perplexity reduction effect caused by the NN LM integration into the SMT system.

	Development corpus			*Test corpus*		
	BLEU	NIST	METEOR	BLEU	NIST	METEOR
baseline	27.00	7.29	53.83	21.77	6.65	49.89
k=10						
3-gram	27.43	7.27	53.82	21.91	6.61	49.62
4-gram	27.54	7.36	54.21	22.26	6.72	50.18
3- and 4-grams mixture	27.34	7.14	53.26	21.54	6.49	49.24
k=12						
3-gram	27.37	7.32	54.10	21.73	6.63	49.68
4-gram	27.47	7.31	53.98	21.91	6.65	49.82
3- and 4-grams mixture	27.47	7.35	54.33	22.07	6.71	50.20

Table 3: Comparative evaluation scores.

system allows gaining up 0.5 BLEU point for the development set and the same value for the test set. The results difference is statistically significant for two of six considred system configurations in terms of BLEU score, while METEOR and NIST metric values vary slightly, never exceeding the statistical significance thresholds.

5. Discussion and conclusions

NN LM shows very quite evident reduction in terms of perplexity (7.5 - 14.8 %), that allows to be beneficial to translation quality for Arabic to English translation task, even considering only the most frequent words from the training corpus. The NN LM was introduced to the n-gram SMT system as a feature function and was used on the reranking step to rescore the N-best list of translation hypotheses.

Of the technique studied, we have found that system configuration providing the better BLEU score correspond to the 4-gram LMs, while the technique of higher and lower order n-gram mixture seems to be promising and in the future we would like to apply this approach to other language pairs and to larger corpora. Surprisingly we have not achieved better system performance moving up the word frequency threshold from 10 to 12, that can be probably explained by high sparseness of the search space.

The idea of correlation of automatic evaluation metrics with the subjective human evaluation metrics assessed in translation quality is introduced in (Paul, 2006): fluency correlates better with BLEU and adequacy correlates best with METEOR, while the NIST metric has only moderate correlation to both subjective human evaluation metrics. Our work demonstrates the potential for NN LM application in the SMT to improve translation fluency, while adequacy keeps invariable.

Acknowledgements

Work partially supported by the Spanish Ministerio de Educación y Ciencia (TIN2006-12767), by the Spanish Government under grant TEC2006-13964-C03 (AVIVAVOZ project), and by the Spanish Government and FEDER under contract TIN2005-08660-C04-02 (EDECAN project).

6. References

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(2):1137–1155.

C. M. Bishop. 1995. *Neural networks for pattern recognition*. Oxford University Press.

M. José Castro and F. Prat. 2003. New Directions in Connectionist Language Modeling. In *Computational Methods in Neural Modeling*, volume 2686 of *LNCS*, pages 598–605. Springer-Verlag.

B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo, and M. Federico. 2005. The ITC-irst SMT system for IWSLT-2005. In *Proceedings of IWSLT 2005*, page 98–104.

J. M. Crego and J. B. Mariño. 2006. Improving statistical mt by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

J. Crego, A. de Gispert, P. Lambert, M. Khalilov, M. Costa-jussà, J. Mariño, R. Banchs, and J.A.R. Fonollosa. 2006. The TALP Ngram-based SMT System for IWSLT 2006. In *Proceedings of IWSLT 2006*, pages 116–122.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceesings of the ARPA Workshop on Human Language Technology (HLT)*, pages 138 – 145.

S. España-Boquera, F. Zamora-Martínez, M.J. Castro-Bleda, and J. Gorbe-Moya. 2007. Efficient BP Algorithms for General Feedforward Neural Networks. In

Bio-inspired Modeling of Cognitive Tasks, volume 4527 of *LNCS*, pages 327–336. Springer.

N. Habash and F. Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 49–52.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP) 2004*, pages 388–395.

J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà. 2006. N-gram based machine translation. *Computational Linguistics*, 32(4):527–549, December.

J.A. Nelder and R. Mead. 1965. A simplex method for function minimization. *The Computer organization*, 7:308–313.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Ann. Meeting of the ACL*, pages 440 – 447, October.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of HLTNAACL04*, pages 161–168.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL 2002*, pages 311–318.

M. Paul. 2006. Overview of the IWSLT 2006 Evaluation Campaign. In *Proceedings of IWSLT06*, pages 1–15.

H. Schwenk, M. R. Costa-jussà, and J. A. R. Fonollosa. 2006a. Continuous space language models for the IWSLT 2006 task. In *Proceedings of IWSLT 2006*, pages 166–173.

H. Schwenk, D. Déchelotte, and J. L. Gauvain. 2006b. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730.

A. Stolcke. 2002. SRILM: an extensible language modeling toolkit. In *Proceedings of the Int. Conf. on Spoken Language Processing*, pages 901–904.

P. Xu and F. Jelinek. 2004. Random forest in language modeling. In *Proceedings of EMNLP 2004*, pages 325–332.

Language modeling for local and Modern Standard Arabic

Ilana Heintz, Chris Brew

Department of Linguistics
222 Oxley Hall
1712 Neil Avenue
Columbus, OH 43210
heintz.38@osu.edu, cbrew@acm.org

Abstract

We propose a Finite State Machine framework for Arabic Language Modeling. The framework provides several decompositions per word based on the forms of possible stems. The statistical modeling is responsible for ranking the most plausible (prefix)-stem-(suffix) sequences higher than the less plausible decompositions. In addition to being useful for Modern Standard Arabic, we show that the framework is easily applied to colloquial Arabic, which suffers from low amounts of text data for use in Natural Language Processing.

1. Introduction

The language modeling task in Arabic faces these three challenges, among others: Lack of textual resources for the dialectal forms of Arabic, a lack of compatibility of tools and resources between Modern Standard Arabic and the locally spoken forms, and the lack of short vowel representations in written forms of either colloquial or Modern Standard Arabic.[1].

It has been shown that one can compensate for the small amount of data available for some Arabic dialects by modeling over morphemes rather than words (Kirchhoff et al., 2006; Choueiter et al., 2006). We propose a finite state machine framework for hypothesizing the morphemes that make up each word. We will show that the FSM framework, which requires minimal definition of the specific word forms within the language, can be used cross-dialectally without loss of functionality. Furthermore, the framework is designed to allow short vowel predictions simultaneously with the decomposition hypotheses.

A number of tools have been developed to determine the morphemes that make up an Arabic word, many of which use finite state tools to do so. For instance, (Habash et al., 2005) present a tool designed to determine the morphological make-up of Arabic words, and include both dialectal and phonological information in their analyses. In (Beesley, 1998), a finite-state framework is proposed that produces highly specific morphological analyses of Arabic words, based on the regularities of how concepts like articles, prepositions, parts of speech, number, gender, and case are expressed. Similarly, (Kiraz, 2000) presents a multi-tape automata approach to morphological analyses of Semitic languages, focusing on fruitfully combining knowledge about roots, patterns, phonological rules, and concatenative morphology.

In these studies and others, the goal is to use as input, or receive as output, a detailed morphological analysis of a term. If the morphological information is given to the FST, the appropriate surface form including all phonologically and orthographically correct segments is expected as output. If a whole word form is given as input, a single analysis of the word's parts is expected. In order to achieve high accuracy, these studies use pre-determined vocabularies of whole words, stems, or roots.

In this study, we are not concerned in particular with the accuracy of the morphological analyses; rather, we are looking for more prolific output from our system. We eliminate the need for pre-determined lists of words, stems, or roots by being highly permissive in the possible analyses. In doing so, we hope to reduce the out-of-vocabulary problem found in automatic speech recognition by giving the language model more options in morpheme types.

In the balance of the article, we describe the form of the finite state machines, the procedure for evaluating our FSM-based language models (LMs), two baseline experiments, and we analyze our results in the context of a potential speech recognition system.

2. Finite State Framework

Our finite state transducers are designed to decompose each Arabic word into all possible (prefix)-stem-(suffix) combinations. We design one finite state transducer for each of 30 possible patterns. One such transducer is shown in Figure 1. These describe the root letter placement and placement of other pattern consonants, such as alif, nun, taa', siin, and miim, that occur in the augmented patterns. In this study, we do not include the short vowel descriptions that are associated with each pattern; however, the design of this framework is such that vowels can be easily inserted in future work. In addition, the transducers allow an affix to occur on either or both sides of the stem.

In the current study, we assume only this pattern information, and we do not restrict the decompositions in any other way. That is, the affixes may be of any length, and may consist of any letters of the alphabet. The characters that fill the root slots are also unrestricted. We could take a more informed view by restricting the alphabet for either the root or affix letters, or restricting the affixes to a certain size. Instead, we allow the frequencies of the resulting morpheme predictions to determine within the LM which are the most likely decompositions.

[1] This work was supported by a student-faculty fellowship from the AFRL/Dayton Area Graduate Studies Insititute, and worked on in partnership with Ray Slyh and Tim Anderson of the Air Force Research Labs

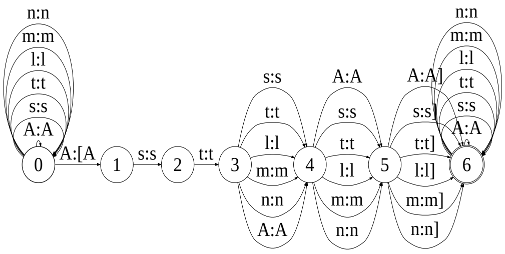


Figure 1: Finite State Transducer for the pattern AstCCC, the pattern for the perfect form of the form X verb, with the example alphabet A,s,t,l,m,n (A represents alif). Affixes may be any length, and can include the same characters as the root and pattern slots. The output will include brackets around the stem portion of the word.

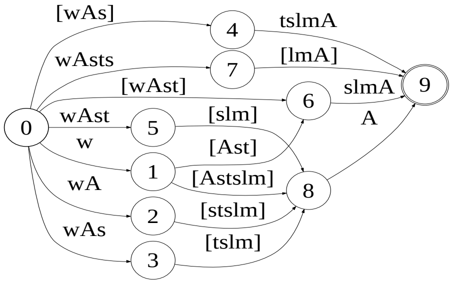


Figure 2: The union of some of the decompositions of the word ‘wAstslmA’, with the possible stems between brackets, and one transition per morpheme. The path labeled 0-1-8-9 is the output of the transducer in Figure 1; the other paths are the output of other patterns’ transducers.

We calculate the union of the 30 pattern FSTs. This set was collected from (Haywood and Nahmad, 1965), taking into account all patterns for the active and passive verbs and for active and passive participles of the ten most common verb paradigms. For this study, the short vowel specifications were removed from the pattern templates, and 30 unique patterns remained. In future studies, we will retain the short vowel information, and apply it to the output of the template transducers.

For each word in the vocabulary, we create a finite state acceptor representing that word, one transition per character, and compose the word acceptor with the pattern transducer. The result is (having performed some additional scripting to augment the AT&T FSM tools), for each word, the union of all possible (prefix)-stem-(suffix) decompositions. This is depicted in Figure 2 for the word *wAstslmA*, “and they (dual) submitted”. In this final representation, each affix and stem are traversed over a single transition, rather than one transition per character. In addition, the stems are identified by surrounding brackets. (In the future, the stems will also include short vowel possibilities, provided by the definitions of the patterns.) Each word is thus represented by a single unweighted morpheme lattice.

3. Procedure

We use two sets of text-only data in our experiments. We will show that the FSM framework that we apply to Modern Standard Arabic (MSA) data can also be successfully applied, with no alteration, to Levantine data, a local form of Arabic spoken in Lebanon, Israel, Jordan, and other places in the Middle East. The MSA data is extracted from the TDT4 Broadcast news corpus (Kong and Graff, 2005), a set of radio and TV news show transcriptions. We use 100 files spanning the duration of the data set and each of the five Arabic sources included in the distribution (Agence France Presse, An-Nahar, Al-Hayat, Voice of America, and Nile TV). These files include 104,757 unique word forms and over 1.8 million total word tokens. We convert the original UTF-8 files into the Buckwalter transcription scheme so as to be more straightforwardly compatible with the AT&T Finite State Toolkit (Mohri et al., 1997), and the SRI language modeling toolkit.

The Levantine data is also distributed through the LDC. We use the QT Levantine Training data sets 3 and 4, as well as the BBN/AUB DARPA Babylon transcripts (Maamouri et al., 2005a; Maamouri et al., 2005b; Makhoul et al., 2005). These consist of transcribed spontaneous speech of people living in the US or Lebanon who speak Levantine Arabic. In this case we use all of the data, which has 61905 unique word forms over a total of 1.75 million words.

The composition of the two corpora bely their differences. Whereas the MSA data contains many very long sentences, the dialogue turns in the Levantine data are quite short, often only one or two words. Only 24% of the Levantine vocabulary is covered in the MSA vocabulary, and only 14% of the MSA words are found in the Levantine word list. Nevertheless, we apply the same language modeling procedure to both data sets.

The procedure outlined below is performed separately for the MSA and Levantine data. That is, we do not include the FSM decompositions of one set in the language modeling or evaluation of the other. Rather than share data between language types, we share the language modeling tool. We perform our experiments using a 10-fold cross-validation scheme for both data sets.

We begin by extracting the vocabulary of each corpus, and individually decomposing each word according to the FSM procedure outlined above. The union of all possible decompositions is stored for each word. If a word does not compose successfully with any of the pattern transducers, then the word is stored as its original acceptor: one transition per character. Next, we transform the training sentences into lattices by replacing each word in the sentence with its decomposed finite state representation, and concatenating the FSMs in the order the words occur in the sentence. We use the SRILM toolkit (Stolcke, 2002) to calculate the n-gram statistics over all of the training sentences. All of the paths and morphemes are equally weighted, so we count the frequency of an n-gram as follows:

$$freq(\text{n-gram}) = \frac{\text{occurrences of n-gram in lattice}}{\text{number of paths in lattice}} \quad (1)$$

Good-Turing smoothing is implemented to allow for unseen words in the test set. We build a 4-gram model for the data. The LM is limited to the 64000 most common morphemes. (This is the maximum size vocabulary for many speech recognizers.)

Having built the LM, we transform the test sentences into morpheme decomposition lattices, as we did with the training sentences. The weights of the LM are applied to the test lattices, and the best path is extracted. We use the score of the best path in our calculation of average negative log probability, which is described fully in Section 5..

4. Comparisons

We compare the FSM-based LMs to two baseline models: a word-based LM, and a morpheme model built using predefined affixes.

We build word-based LMs for each of the ten-fold cross-validation training sets for both Levantine and MSA. Again, we use Good-Turing smoothing and 4-grams.

For the affix model, we begin with the pre-defined set of MSA affixes given in (Xiang et al., 2006). For each word in the vocabulary, we determine one or no (prefix)-stem-(suffix) decompositions, based on the presence of any of the affixes, or sequences of affixes, and the length of the remaining stem. Our heuristics are designed to separate the largest possible affix while assuring that the stem is at least 3 characters long. We transform the training data by replacing each word with its morpheme representation, where morphemes are separated by whitespace. We build the LM over the decomposed training data. Because the decompositions are determined *a priori*, we can transform the test data in the same way, and evaluate the LMs on morpheme data. This baseline is a simplified approximation of the methods used in (Xiang et al., 2006) and (Choueiter et al., 2006).

For the Levantine data, we have two affix models. In the first, we apply the same MSA affixes to the Levantine data. In the second, we define a new set of affixes based on information in a textbook on spoken Levantine (Hussein, 1993). The same procedure as above is applied.

5. Evaluation

In order to evaluate the FSM-based LM against the word-based and affix-based models, we use an adapted measure of perplexity introduced in (Kirchhoff et al., 2006), called average negative log probability. Perplexity measures the average branching factor of the model, and is calculated as:

$$PP(x_1 \ldots x_n) = 2^{\frac{1}{n}\sum_{x_i=4}^{n} logP(x_i|x_{i-1}^{i-3})} \qquad (2)$$

where n is the number of items in the test set. A lower perplexity indicates a stronger model. However, calculating this measure for each of our models would lend bias towards the FSM and affix models, as n, the number of morphemes, will invariably be larger for those models. Instead, we calculate the average negative log probability as follows:

$$PP(x_1 \ldots x_n) = \frac{1}{n}\sum_{x_i=4}^{n} logP(x_i|x_{i-1}^{i-3}) \qquad (3)$$

where n is the number of $words$ in the test set, as determined by the word model. This allows for a more fair comparison among the models.

	Word	Affix	FSM
Avg Neg Log Prob	4.61	5.17	4.82
Unigram	96.96	99.37	99.10
Bigram	15.75	50.72	64.36
Trigram	0.67	9.61	24.22
4-gram	0.12	2.14	8.62

Table 1: Average negative log probability and n-gram coverage (%) results for the MSA test set.

	Word	Affix-Lev	Affix-MSA	FSM
Avg Neg Log Prob	4.35	5.71	5.68	3.45
Unigram	91.30	77.26	78.77	98.98
Bigram	11.19	10.18	10.76	68.24
Trigram	0.26	0.29	0.29	31.01
4-gram	0.00	0.0	0.0	10.53

Table 2: Average negative log probability and n-gram coverage (%) results for the Levantine test set.

In addition, we calculate the coverage of each size n-gram for each model. For the word and affix models, we simply count how many of the n-gram tokens that appear in the test set are included in the language model. For the FSM model, we calculate the coverage of the language model over the morphemes that appear in the best path of the test set lattices. In addition, we compare the word and FSM-based models by calculating the percentage of word n-grams in the word model exactly accounted for by morpheme n-grams in the FSM-based LMs.

6. Results

In Tables 1[2] and 2, the results are shown for one fold that represents the pattern found across all folds of the cross-validation scheme. The results in the first row show that the FSM model is a good model of the language. That is, in both cases the average negative log probability of the FSM model is lower than that the affix model. The score of the FSM-based model is lower than the word model in the Levantine case, and very close to the word model in the MSA data set. It is difficult to judge significance for these measures, so it is not appropriate to claim that the FSM model outperforms the other two. The best way to test that claim is to implement the models in an ASR system. These results do show, however, that the FSM model is a promising method for Arabic language modeling. This is especially true in that the same method, with the same background knowledge of standard Arabic morphology, worked well for both Modern Standard Arabic and Levantine Arabic.

The coverage statistics in Tables 1 and 2 confirm that the FSM model is a useful method. Coverage of all n-grams is greatest for the FSM model for both languages. This may be because the FSM model results in many one- or two-

[2] These values differ from those in (Heintz, 2008). In this study, we do not use a single unknown symbol for all OOV terms. Also, we do not include character-wise FSMs in the decomposed representation of each word.

	MSA	Levantine
Unigram coverage (%)	35.84	30.35
Bigram coverage (%)	12.58	43.94

Table 3: Percentage of word unigrams and bigrams covered by the FSM model for each of the two data sets.

	MSA	Lev
word-25%, fsm-75%	5.06	3.33
word-50%, fsm-50%	5.29	3.47
word-75%, fsm-25%	5.70	3.69

Table 4: Average negative log probability results of interpolating the word and fsm models for both Levantine Arabic and MSA, with three interpolation weighting schemes. Results represent a single fold of the cross-validation.

character morphemes. These come from both the affixes and the words that do not compose with the patterns, and are stored as one-character-per-transition acceptors. These morphemes are likely to repeat in both training and test sets; however, they are harder to distinguish in an ASR task.

In addition, we explore how well the morpheme n-grams stored in the FSM-LM cover the word unigrams and bigrams stored in the word-based LM. These results are shown in Table 3. Surprisingly, the coverage is not very good. It seems that the n-grams stored in the FSM-based model cover more inter-word sequences than whole words. Again, this may have to do with the tendency of the FSM model towards short morphemes, which recur often, as opposed to the stems, which are less likely to repeat. One way to amend this would be to store a whole-word morpheme representation for words that do not decompose, rather than storing a character-based morpheme representation.

To test whether interpolation of the word- and FSM-based models is beneficial, we must adjust the FSM representation slightly. We combine the word and morpheme representations by creating a single-transition FSA for each word, and calculating the union of that FSA with the morpheme FST defined earlier for that word. We use these word + morpheme lattices to build the test sentence lattices for the interpolated models. The results for three interpolation weighting schemes are shown in Table 4.

We see that for both MSA and Levantine data, using a heavier weight for the FSM part of the interpolation is best. For the MSA data, the scores for the FSM model and word model shown in Table 1 are lower than any of the interpolated models. For the Levantine data, comparing to Table 2, interpolating the word model into the FSM model, giving more weight to the FSM model, achieves a better score than the FSM model alone. It seems that the extra context gained by interpolating the word model is beneficial for the type of data found in the Levantine corpus, but the same is not true for the more varied data found in the MSA corpus.

7. Conclusion & Future Work

Thus we find that the FSM framework, originally built to work with MSA data, extends to Levantine Arabic. We find that proposing affixes based on probable stem forms works as well as defining the affixes in advance to search out stems. This method works in large part because the language model statistics are able to rule out the least likely decompositions. Implausible affixes are bound to be less frequent in this model than verifiable affixes.

In future work, we will interpolate this morpheme model with a stem-only model, to see if the increased scope of the stem n-grams can improve the model. Applying the language model weights to the individual word lattices, then rebuilding the model based on weighted decompositions in an E-M process may prove to be beneficial. In addition, we plan to implement these language models in a speech recognition system.

8. References

Kenneth R Beesley. 1998. Arabic morphology using only finite-state operations. In Michael Rosner, editor, *Computational Approaches to Semitic Languages: Proceedings of the Workshop*, pages 50–57. Montréal, Québec.

Ghinwa Choueiter, Daniel Povey, Stanley F. Chen, and Geoffrey Zweig. 2006. Morpheme-based language modeling for Arabic LVCSR. In *ICASSP 06*, pages 1053–1056.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proc. ACL Wrkshp on Comp'l Approaches to Semitic Languages*, pages 17–24.

J.A. Haywood and H.M. Nahmad. 1965. *A New Arabic Grammar of the Written Language*. Lund Humphries, Burlington, VT.

Ilana Heintz. 2008. Arabic language modeling with finite state transducers. In *ACL 2008*, Columbus, OH.

Lufti Hussein. 1993. *Levantine Arabic for Non-Natives*. Yale University Press, New Haven, CT.

G.A. Kiraz. 2000. Multi-tiered nonlinear morphology using multi-tape finite automata: A case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105.

Katrin Kirchhoff, Dimitra Vergyri, Kevin Duh, Jeff Bilmes, and Andreas Stolcke. 2006. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech and Language*, 20(4):589–608.

Junbo Kong and David Graff. 2005. TDT4 multilingual broadcast news speech corpus.

Mohamed Maamouri, Tim Buckwalter, and Hubert Jin. 2005a. Arabic cts levantine fisher training data set 3, transcripts.

Mohamed Maamouri, Tim Buckwalter, and Hubert Jin. 2005b. Levantine arabic qt training data set 4 (speech + transcripts).

John Makhoul, Bushra Zawaydeh, Frederick Choi, and David Stallard. 2005. Bbn/aub darpa babylon levantine arabic speech and transcripts.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 1997. At&t FSM Library.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado.

Bing Xiang, Kham Nguyen, Long Nguyen, Richard Schwartz, and John Makhoul. 2006. Morphological decomposition for Arabic broadcast news transcription. In *Proc. ICASSP 2006*, pages 1089–1092.

Towards a syntactic lexicon of Arabic verbs

Noureddine LOUKIL, Kais HADDAR, Abdelmajid BEN HAMADOU
Institut Supérieur d’Informatique et Multimédia de Sfax,
BP 1030, Sfax 3002, Tunisie
E-mail: {noureddine.loukil,
Abdelmajid.benhamadou}@isimsf.rnu.tn,
Kais.haddar@fss.rnu.tn

Abstract

This paper presents the modelling of an extensional syntactic lexicon for verbal entities in Arabic, based on the initiative for lexical resources normalisation LMF (Lexical Markup Framework). The specific syntactic behaviors for verbs in Arabic language are identified and presented with examples. Each verbal entry is specified with a list of accepted syntactic patterns describing the set of accepted arguments with their different constraints. The syntactic extension of LMF and the XML structure of lexical entries are presented in accordance with the LMF normative initiative. This lexicon would be very useful for NLP community because it enables comfortable use in applications due to its normalised representation.

1. Introduction

The majority of natural language processing applications depend on computational lexicons to retrieve the linguistic knowledge needed to achieve their functionality. Computational lexicons are thus necessary resources and their coverage and quality affect the results of applications using them. However, the construction of such resources with the required quality and coverage demands considerable amount of time and effort. Furthermore, computational lexicons that encode specific linguistic information, like syntactic lexicons, are more and more difficult to build. This is generally due to the absence of dictionaries or existing lexical databases from which we can extract the required specific knowledge. This kind of resources is normally unavailable for less studied languages like Arabic.

To our knowledge, there is no such a specific resource for Arabic but there are morph syntactic lexical databases that partially integrate syntactic information (khemakhem2006). Many other works addressed the description of Arabic verbs syntax within unification grammars lexicons (Loukil2007) and the standardization of lexical databases representation (Loukil2006).
In this paper, we present our ongoing effort to build a valence lexicon for Arabic and emphasizes on the description of syntactic behavior of verbal entities. Section 2 describes the linguistic content and the logical structure of the lexicon while section 3 details the encoding of the various kinds of information within verbal entries. Finally, section 4 presents a discussion of main issues and perspectives.

2. Linguistic Content of the lexicon

The Arabic language is a Semitic language characterized by two major phenomena: agglutination and non-vocalization. Agglutination makes it difficult to process texts and creates a necessity to lemmatization. A big number of studies have examined this issue, and proposed methods and algorithms to lemmatize Arabic texts (dichy2001). Nevertheless, those methods have to be implemented in any NLP system in order to cope with this problem. An extensional lexicon is a very practical solution to that situation (Akrout2005). Describing all inflections of a given word, say a verb, is linguistically consistent if the lexicon states all the morphological and syntactic relations between them. The lexicon we are presenting in this paper is extensional: each verbal form is described by an independent entry.

All Arabic verbal forms, both primitive and derivative, have two voices, the active and the passive with the exception of intransitive verb. Those verbs designate not an act but a state or a condition, i.e. (صَلُحَ) (to be good, right, in order). There are just two temporal forms: the perfect expressing a finished act and the imperfect used for acts that are just commencing or in progress. Arabic verbs have five moods: indicative, subjunctive, jussive or conditional, imperative and energetic.

The linguistic information encoded in the lexicon consists of a list of all possible syntactic frames for each verbal entry together with a set of constraints associated either to the verb or to the arguments.

2.1. Arabic language resources

The major source of verbal syntactic information is (gigano2004). This printed lexicon contains more than 5000 transitive verbal entries with their complementation conjunctive particles, along with explanations and examples. Although this source is incomplete, it has proven to be a valuable starting point for studying syntactic behaviors of Arabic verbs. The second source we used for extracting verbal constructions consists of the major printed Arabic dictionaries, all of them available in computer-readable format: “Lesan Al-Arab” (The Arabic tongue), “Al-ain” (The source) and “Al-mouhit” (The Ocean). These resources are relatively precise and complete but need processing to extract the syntactic information dispersed in main entries and examples.

2.2. Organization of the linguistic knowledge

The linguistic knowledge has a very complex structure that must be formalized. This formalization has to be complete and precise. In this work, we tried to follow the most commonly used practices in syntax description and especially those formulated by the lexical markup framework (LMF_revision_14) as a syntactic extension. This extension is a generic model for syntax description and it is not intended for grammar description. Thus, we used those guidelines to design the organization of our lexicon.

Each lexical entry, corresponding to a verbal entry in our case, has a set of syntactic behaviors. Each of them is actually described by a list of appropriate sub categorization frames that represent syntactic constructions.

A sub categorization frame consists of an ordered list of the arguments required by the verbal entry, and a set of constraints on arguments. Thus, we describe, firstly, the type and the number of the valence arguments, and secondly, the types of constraints associated with arguments. Table 1 gives the complete list of verbal syntactic behavior.

The first syntactic behavior is for intransitive verbs (IV) as in (a). The second syntactic behavior concerns verbs that require just one direct complement, as in (b). The fourth describes verbs requiring one complement introduced by a particle.

Example	**Verbal Syntactic Behavior**
a	Intransitive verb (IV)
b	Transitive verb requiring one direct complement (TVD)
c	Transitive verb requiring two direct complements (TVDD)
d	Transitive verb requiring three direct complements (TVDDD)
e	Transitive verb requiring one complement with particle (TVP)
f	Transitive verb requiring two complements with particles (TVPP)
g	Transitive verb requiring one complement either direct or by particle (TVX)
h	Transitive verb with two complements: the first is direct and the second is with particle (TVDP)

Table 1: verbal syntactic behaviors.

(a)

نام الطفل

The child slept

(b)

كَسر الطفل القلم

The child broke the pen

(c)

أعطى التلميذ درسا

He gives the student a lesson

(d)

أنبأ الأستاذ تلاميذه الأمر سهلا

The teacher informs his students the matter was easy

(e)

ذهب إلى المدرسة

He goes to the school

(f)

دعا لأخيه بالنجاح

He pray for his brother to succeed

(g)

تحرك نحوه / في اتجاهه

He moves (*him/ to his direction)

(h)

اختار تلميذا من الصف

He chooses a student from the class

2.2.1 Valence arguments

In our lexicon, each verb has a unique syntactic behavior chosen among the patterns of table 1. Each syntactic behavior contains two sub categorization frame sets. The first states the possible sub categorization frames in verbal sentences. The second states the possible sub categorization frames in nominal sentences.
Each sub categorization frame describes the required verbal arguments. We deal with active valence and we do not cope with modifiers that can occupy the same place of required arguments without eliminating them.

A verb subcategorize for one, two or three complements. The complement may be direct, i.e. a simple noun, or introduced by a conjunctive particle from the finite set {في، عن،اللام ، على، الباء، إلى، من}. Each verb selects his accepted conjunctive particles in the case it is transitive by particle. For instance the verb حصلَ can accept the particles {من،اللام على، الباء}.

2.2.2 Constraints

A Constraint is defined as a condition that must be satisfied for the verb or one of its syntactic arguments. Verbs, which are transitive by particle, selects for the particles that can introduce each argument. This kind of constraints applies to arguments.

نظر إلى الكتاب

He looks to the book

نظر في الأمر

He examined the issue

نظر على الأمر *

He examined on the issue *

The verb “نَظَرَ” (to see), accepts arguments introduced by the conjunctives إلى and في but does not accept those introduced by the conjunctive على. This constraint is described by the “restriction” property which states selected conjunctives for a particular verb.

3. Structure of verbal entries

We are interested in a current work led by the ISO TC 37/SC committee within the LMF (Francopoulo2005) project which aims the elaboration of the future standard ISO 24613. This standard is mainly based on work carried within the projects EAGLES (Standards1996), MULTEXT (Calzolari1996) and GENELEX (Antoni-Lay1994), which are initiatives to standardize linguistic information representation. LMF propose an abstract representation meta-model that can be instantiated to build new multi-level lexical resources.

The structure of each verbal entry follows the guidelines described in the Lexical Markup Framework (ISO TC 37/SC 4). The final encoding format is an XML based description. Every verbal inflexion is represented by one lexical entry that contains a morphology specification (Form), a syntactic specification (Syntactic Behavior) and a semantic specification (Sense).

The syntactic extension is built around the Syntactic Behavior component. A syntactic behavior is a syntactic formation pattern which may be used by several lexical entries to capture syntactic redundancy within the lexicon. A syntactic behavior is described by the set of permitted syntactic formations eventually grouped in semantically disjoint subsets. A frame represents the set of possible syntactic constructions associated to a predicate and actually realized by the combination of several complements or positions. Within a frame, possible instances of the positions can lead to syntactically correct phrases. In other words, a frame can be seen as valence pattern providing a specification about the order and the nature of permitted positions instances. A lexical entry may have several frames providing each several mandatory or optional positions. Each position proposes possible realizations and their morphosyntactic descriptions given within a SyntacticActant component. The Self component describes the morpho syntactic of the actual lexical entry.

A syntactic frame is a specification of a possible syntactic construction. Each frame has an identifier, so it can be used by more than one syntactic behavior and, thus, by more than one verbal entry. The frame node is composed of three elements, the verb itself, an ordered list of syntactic slots, and an example of the verb used with the given construction. We represent below one syntactic behavior of the verb "تَحَرَّكَ" (to move).

```
<SyntacticBehavior id="TVX" label=”Transitive Verb with Direct or
Particle Complement”>
        <SubcategorizationFrameSet id=”TVD” label=”frames for
verbal sentences”>
                <SubcategorizationFrame id="TVD" label=”Transitive
Verb with Direct complement in verbal sentences”
                        example=”تحرّك صوبه”>
                        <LexemeProperty position=”1"
partOfSpeech=”verb” mood=”indicative” voice=”active” />
  <SyntacticArgument function="subject" syntacticConstituent=”NP" />
                        <SyntacticArgument function="object"
syntacticConstituent=”NP" />
</SubcategorizationFrame>
                <SubcategorizationFrame id="TVP" label=”Transitive
Verb with Particle complement in verbal sentences”
                        example=” تحرّك في اتجاهه ”>
                        <LexemeProperty position=”1"
partOfSpeech=”verb” mood=”indicative” voice=”active” />
                        <SyntacticArgument function="subject"
syntacticConstituent=”NP" />
                        <SyntacticArgument function="object"
syntacticConstituent=”PP"

        Introducer=”Particle” restriction=”إلى، في، على، الباء، من، اللام”/>
                </SubcategorizationFrame>
        </SubcategorizationFrameSet>

         <SubcategorizationFrameSet id=”TVDn” label=”frames for
nominal sentences”>
                <SubcategorizationFrame id="TVDn"
label=”Transitive Verb with Direct complement in nominal sentences”
                        example=”القوّات تحرّكت نحو العدوّ”>
                        <LexemeProperty position=”2"
partOfSpeech=”verb” mood=”indicative” voice=”active” />
                        <SyntacticArgument function="topic"
syntacticConstituent=”NP" />
                        <SyntacticArgument function="object"
syntacticConstituent=”NP" />
                </SubcategorizationFrame>
                <SubcategorizationFrame id="TVPn" label=”Transitive
Verb with Particle complement in nominal sentences”
                        example=”القوّات تحرّكت إلى الأمام”>
                        <LexemeProperty position=”2"
partOfSpeech=”verb” mood=”indicative” voice=”active” />
                        <SyntacticArgument function="topic"
syntacticConstituent=”NP" />
                        <SyntacticArgument function="object"
syntacticConstituent=”PP"

        Introducer=”Particle” restriction=”إلى، في، على، الباء، من، اللام”/>
```

The verb node "LexemeProperty" encodes grammatical constraints, for instance, the tense and the mood. After which a list of "SyntacticArgument" nodes is provided. Constraints on each syntactic argument are coded within his specific "syntacticArgument" node by assigning the appropriate values to syntactic features.

4. Conclusion

The lexicon is based on an extensional model and, thus, we are not constrained to deal with the complex morphology/syntax relation. We are working on developing techniques for feeding the lexicon with other corpora sources like the web. In this paper, we presented a syntactic lexicon of Arabic verbal entities. Each verbal entry is specified with a list of accepted syntactic frames describing the set of possible arguments with their different constraints along with examples from printed sources. The encoding format is based on a future normative proposition to ensure the interchangeability and reuse of the resource. The lexicon may be interesting for different NLP applications.

5. References

(Loukil2007) Loukil N., Haddar K., Benhamadou A., (LGC 2007), « Une proposition de représentation normalisée des lexiques des grammaires d'unification » 26th Colloque International sur le Lexique et la grammaire, Bonifacio, Corse, France.

(Loukil2006) Loukil N., (RECITAL2006), « Une proposition de représentation normalisée des lexiques des grammaires d'unification » Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues, Leuven. Belgique.

(Akroou2005) AKROUT A., Modélisation d'un lexique flexionnel. Application à l'arabe classique.

Master's thesis, Université de Metz, France.

(Antoni-Lay1994)Antoni-Lay, MH., Francopoulo G. and Zaysser L. A generic model for reusable lexicons: The genelex project. Literary and Linguistic Computing.

(Calzolari1996) Calzolari , N. M. M. Multext - Common Specifications and Notation for Lexicon Encoding. Rapport interne.

(Standards1996) EAGLES, Reports of the Computational Lexicons Working Group. Internal Report.

(gigano2004) Gigano, A., The Lexicon of verbal complementation in Arabic language, Dar Elrateb, edition 2004.

(khemakhem2006), ArabicLDB : une base lexicale normalisée pour la langue arabe

Master's thesis, University of Sfax, Tunisia.

(Dichy2001), "Une première classification des verbes arabes en fonction de leur structure d'arguments", Génération Systématique de la langue et Traduction automatique, special issue of Recherches Linguistiques/Linguistic Research, Rabat, IERA

Automatic Morphological Rule Induction for Arabic

Ahmad Hany Hossny
Faculty of Computers and Information
Cairo University
ahmad.hossny@gmail.com

Khaled Shaalan
Faculty of Informatics
The British University in Dubai,
PO Box 502216, Dubai, UAE
khaled.shaalan@buid.ac.ae

Aly Fahmy
Faculty of Computers and Information
Cairo University
A.fahmy@fci-cu.edu.eg

Abstract

In this paper, we introduce an algorithm for morphological rule induction using meta-rules for Arabic morphology based on inductive logic programming. The processing resources are a set of example pairs (stem and inflected form) with their feature vectors, either positive or negative, and the linguistic background knowledge from the Arabic morphological analysis domain. Each example pair has two words to be analyzed vocally into consonants and vowels. The algorithm applies two levels of mapping: between the vocal representation of the two words (stem, morphed) and between their feature vector. It differentiates between both mappings in order to accurately deduce which changes in the word structure led to which changes in its features. The paper also addresses the irregularity, productivity and model consistency issues. We have developed an Arabic morphological rule induction system (AMRIS). Successful evaluation has been performed and showed that the system performance results achieved were satisfactory.

1. Introduction

Many researchers have attacked the morphological rule induction problem in a variety of languages but only a few limited researches have focused on Arabic language. This is due to the characteristics and peculiarities of Arabic language, lack of resources, and the limited amount of progress made in Arabic natural language processing in general.

The following example gives a general idea about how rules are mapped by our morphological rule induction system. Consider the weak verb “وقى” (to-protect) which has the vocal representation “VCV”. In present tense the word has the form “يقى” which is produced by removing the radical character “و” and adding the present tense prefix character “ي”. The mapping rule for these two words is as follows:

V2C1 ي (tense:present) → V2C1و (tense:past)

Applying this rule to the past tense weak verb "وعى"(to-be conscious, past tense), which has the vocal representation (VCو), yields the present tense form “يعى”. The rule induction algorithm applies two levels of mapping: between the vocal representation of the two words (stem, morphed) and between their feature vector. It differentiates between both mappings in order to accurately deduce which changes in the word structure led to which changes in its features.

This paper advances the state of the art in the Arabic morphological rule induction by: 1) utilizing the vocal properties in the analysis of Arabic words (Beesley, 1996) for handling irregularities, 2) adopting morphological rule induction techniques in order to handle productivity and consistency issues, and 3) adding some semantic features to the feature vector of Arabic words in order to be convenient for a wide spectrum of natural language processing systems such as machine translation and search engines.

The rest of this paper is organized as follows. Section 2, presents different linguistic acquisition techniques. In Section 3, we introduce the proposed example-based Arabic morphological rules acquisition. Experimental results are discussed in Section 4. In Section 5, concluding remarks and directions for future work are derived.

2. Related Work

Grammar acquisition or grammar induction is usually done at the syntactic level for most Latin-based languages as they usually do not have great deal of irregularities (Daille, 2000). On the other hand Semitic languages such as Arabic and Hebrew need a significant handling at the morphological level and sometimes at the morphosyntactic level. In the following, we briefly present the major induction techniques.

2.1 State Space Generation

It is based on trial and error where all possible grammar rules are generated using all possible combination of features (Miller & Fox, 1994). This gives a full coverage of all rules but gives also a massive amount of ambiguities which reduces the accuracy of the parsing process and its efficiency as well.

2.2 Slots and Filler Technique

It is based on template matching of the sentence structure highly needed in lots of NLP applications such as machine translation (Edelman, Solan, Horn, Ruppin, Rich, 2003).

2.3 Stochastic Techniques

2.3.1 Genetic Programming (Pappa & Freitas, 2004) utilizes prior knowledge of rule induction domain to build asset of functions and terminals used to evolve a rule and then compute fitness value of candidate rule induction algorithms of genetic programming population.

2.3.2 Structural Zeros technique uses statistical or machine learning techniques to cover also any an unobserved combination that may arise due to sparse data or hard syntactic constraints (Mohri & Roark, 2006).

2.4 Case-Based Reasoning

It employs relevance weighting to access similarities between cases, making use of rule induction results to assign weights to each attribute-value pair of the query case (Cercone, An, Chan, 1999). Cases in the case base can then be ranked according to their probability of relevance to the new case.

2.5 Neural Networks

It treats the rule induction process as a classification problem aims to classify the sample to some rules, so it propose an activation function that simulates the behavior of logic induction, such neural system is to be trained using input data and output classification of the data to build the rule on the neurons basis which is kind of supervised learning for when it gets the relevant weights as basis for neurons it constructs the rules (Silva & Ludermir, 1999).

Most of the mentioned techniques need lots of data samples to build the language model especially in the statistical and stochastic domains, which is not the case in the problem we handle where we need to build logical language model with few examples in a decreasing growing rate.

3. Example-Based Arabic Morphological Rules Acquisition

In this section an automated morphological acquisition algorithm is presented along with its influence on the computational morphology module, it applies the inductive logic programming (Muggleton, 1999) concepts by dependency on example pairs and logic behind the acquisition process which is similar to automatic word guessing (Mikheev, 1997) that is used in French rule induction (Daille, 2000). In our discussion, we start with giving some related definitions of different sets, ordered pairs and operators. This is followed by the specification of the proposed algorithm of the automated rule generation. Finally, the parsing algorithm is presented.

The proposed learning technique depends on a set of example pairs and their corresponding feature vectors that are used for the training process.

Definition 3.1: Let $\Sigma = \{\text{أ}, \dots, \text{ي}\}$ be the set of Arabic literals. Then a set Σ^+ is the set of Arabic words of at least one literal.

Definition 3.2: Let Σ^+ be the set of Arabic words of at least one literal. Then the binary operator $+: \Sigma^+ \times \Sigma^+ \rightarrow \Sigma^+$ is called the suffix and prefix concatenation operator.

Definition 3.3: Let Σ^+ be the set of Arabic words of at least one literal. Then the binary operator $*: \Sigma^+ \times \Sigma^+ \rightarrow \Sigma^+$ is called the infix concatenation operator.

Definition 3.4: Let Σ^+ be the set of Arabic words of at least one literal. Then the binary operator $\dot{-}: \Sigma^+ \times \Sigma^+ \rightarrow {\Sigma^+}^4$ is called the affix separation operator. It compares two Arabic words and maps the differences into a prefix, a suffix, an infix, and a stem word. The set ${\Sigma^+}^4$ is then called the set of uninflected words.

Definition 3.5: Let $\Sigma_V = \{\mathrm{c}, \mathrm{v}\}$ be the set of vocal literals, where v denotes to vowel literal and c is the notation for consonant literals. Then the set Σ_V^+ is the set of vocal words of at least one literal.

Definition 3.6: Let Σ_V^+ be the set of vocal words and Σ^+ be the set of Arabic words of at least on literal. Then a unary function $\Upsilon: \Sigma^+ \rightarrow \Sigma_V^+$ be a mapping function from Arabic words to vocal words.

Definition 3.7: Let $F = \{f_i | 1 \leq i \leq n\}$ be the feature set of an Arabic word. Let V_i be the value set of the word feature f_i. Then the ordered pair $(f_i, v) \in \{f_i\} \times V_i$ is called a feature-value pair and the set $P_{f_i}^{V_i}\Big|_{i=1,\dots,n}$ is the set of feature-value pairs.

Definition 3.8: Let $P_{f_i}^{V_i}\Big|_{i=1,\dots,n}$ be the set of feature-value pairs. An ordered tuple

$$\left(p_{f_1:v_1}, \dots, p_{f_n:v_n}\right) \in \prod_{i \in \{1,\dots,n\}} P_{f_i}^{V_i}$$

is called a featured vector.

Definition 3.9: Let $P_{f_i}^{V_i}\Big|_{i=1,\dots,n}$ be the set of feature-value pairs. Let ${\Sigma^+}^4$ be the set of uninflected words. A unary function $\psi: {\Sigma^+}^4 \rightarrow \prod_{i \in \{1,\dots,n\}} P_{f_i}^{V_i}$ is called a featured vector mapping function.

A featured word is composed of sequence of vocal characters and a feature vector. It works both ways. It is a data structure while generating the rules and a primary key in the parsing process.

Definition 3.10: Let $\prod_{i \in \{1,\dots,n\}} P_{f_i}^{V_i}$ be the set of feature vectors. Let Σ_V^+ be the set of vocal words. Then a set

$$W^F = \Sigma_V^+ \times \prod_{i \in \{1,\dots,n\}} P_{f_i}^{V_i}$$

is called the set of featured words and an ordered pair $w^F \in W^F$ is called a featured word.

The distance will be measured by considering the sequence of vocal word. There are three cases when measuring a distance depending on the examined vocals whether they are similar vocal words, semi different vocal words or totally different vocal words. In similar vocal words, both words being examined have the same sequence of consonants and vocals.

Consider the word ’يلعبون’ (they-are-playing [masculine]) having the vocal representation ‘ونC2C1C0ي’. It vocally matches the word ‘يشربون’ (they-are-drinking [masculine]) and hence has the same vocal representation. Semi-different vocal words are having different vocal representation. They only differ in consonants.

```
Algorithm: derive_rule
Inputs: stem $x$ and morphed word $\mathbf{x}$
Ouputs: $R$

1.  Extract the stem $x$ from the morphed word $\mathbf{x}$ to get the
    uninflected word $\tilde{x}$ and the affixes that form the prefix(es) $\mathbf{x}^p$,
    infix(es) $\mathbf{x}^i$ and suffix(es) $\mathbf{x}^s$

        $(\tilde{x}, \mathbf{x}^p, \mathbf{x}^i, \mathbf{x}^s) \leftarrow \mathbf{x} \dot{-} x$

2.  Derive the feature vectors for both $\mathbf{x}$ and $\tilde{x}$

        $(\mathbf{x}_{f_1:v_1}, \dots, \mathbf{x}_{f_n:v_n}) \leftarrow \psi(\mathbf{x})$
        $(\tilde{x}_{f_1:v_1}, \dots, \tilde{x}_{f_n:v_n}) \leftarrow \psi(\tilde{x})$

3.  Differentiate and compare the feature vectors of  both of $\psi(\mathbf{x})$
    and $\psi(\tilde{x})$ word via unification;

        $(\mathbf{x}_{f_1:v_1}, \dots, \mathbf{x}_{f_n:v_n}) \doteq (\tilde{x}_{f_1:v_1}, \dots, \tilde{x}_{f_n:v_n})$

    If $\mathbf{x}_{f_i:v_i} = \_$ and $\tilde{x}_{f_i:v_i} = v_i$ then
        $r: r^L_{f_i:\_} \rightarrow \mathbf{x}^p + r^R_{f_i:v_i} + \mathbf{x}^s$
    If $\mathbf{x}_{f_i:v_i} = \mathrm{v_i}$ and $\tilde{x}_{f_i:v_i} = \_$ then
        $r: r^L_{f_i:v_i} \rightarrow \mathbf{x}^p + r^R_{f_i:\_} + \mathbf{x}^s$
    If $\mathbf{x}_{f_i:v_i} = \mathrm{v_i}$ and $\tilde{x}_{f_i:v_i} = \mathrm{v_i}$ then
        $r: r^L_{f_i:V_i} \rightarrow \mathbf{x}^p + r^R_{f_i:V_i} + \mathbf{x}^s$
    If $\mathbf{x}_{f_i:v_i} = \mathrm{v_i}$ and $\tilde{x}_{f_i:v_i} = \tilde{\mathrm{v}}_\mathrm{i}$ then
        $r: r^L_{f_i:v_i} \rightarrow \mathbf{x}^p + r^R_{f_i:\tilde{\mathrm{v}}_\mathrm{i}} + \mathbf{x}^s$

4.  Generate vocal representations of $\mathbf{x}$.

        $\mathbf{x}^\Upsilon \leftarrow \mathbf{x}^p + \Upsilon(\mathbf{x}) + \mathbf{x}^s$

5.  Generate the $j^{\mathrm{th}}$ rule $\mathrm{r_j}$

        $R: (\mathbf{x}^\Upsilon, r^L_{f_i:V_i}) \rightarrow \mathbf{x}^p + r^R_{f_i:\tilde{V}_\mathrm{i}} + \mathbf{x}^s$

    where $1 \le i \le n$, and $n$ is the number of features.

6.  Return $R$.
```

Figure 1: Pseudo code Algorithm for learning Arabic morphological rules from examples

Consider the word ‘يشددون’ (they-are-stressing-on [masculine]) with a vocal representation ‘ونC2C1C0ي’. It has one more consonant than ‘يشدون’ (they-are-pulling [masculine]) with a vocal representation ‘ونC1C0ي’. The last cases are totally different vocal words where vowels are misplaced or differ in number such as ‘ينوون’ (they-intending [masculine]) which has the vocal representation ‘ونV1C0ي’ which having past ‘نوى’ which has the vocal representation ‘ىV1C0’ and ‘يهوون’ (they-interested in [masculine]) which has the vocal representation ‘ونV1C0ي’ which having past ‘هوى’ which has the vocal representation ‘ىV1C0’.

The rule generation algorithm listed below in Figure 1 describes the steps followed to get the literal and feature vectors mapping between a stem and its inflected form. Throughout this algorithm, feature vectors are represented as lists. The algorithm uses a unification-based assignment. The underscore symbol ‘_’ represents a ‘do not care’ case.

The algorithm takes as input a stem $\tilde{x} \in \Sigma^+$ and an inflected word form $\mathbf{x} \in \Sigma^+$. Then, it applies the n-gram algorithm for computing the dissimilarity between the two input words to derive the breakdowns of the inflected word into affixes and a stem $\tilde{x} \in \Sigma^+$. Then it compares the stem $\tilde{x}$ with the inflected word $\mathbf{x}$ to derive the literal mapping rules. The feature vectors of both the input word $(\mathbf{x}_{f_1:v_1}, \dots, \mathbf{x}_{f_n:v_n})$ and the stem $(\tilde{x}_{f_1:v_1}, \dots, \tilde{x}_{f_n:v_n})$ are also compared to derive the features mapping rules. Finally, a rule $\mathrm{r_j}$ that represents both the literal and features mapping is generated.

Every generated rule R is verified against an indexed rule base for redundancy, ambiguity and transitivity before it is accepted to be added to the rule base.

The following techniques are applied in order to maintain an efficient rule base:

- If the RHS part of a newly generated rule has included an expression that occur as a whole RHS of another existing in the rule base then replace this expression by the LHS of the existing rule. For example, give the following;

$$R_i: (\mathbf{x}_i^\Upsilon, r^L_{f_i:v_i}) \rightarrow \mathbf{x}^p + r^R_{f_i:\tilde{v}_\mathrm{i}} + \mathbf{x}^s$$

$$R_j: (\mathbf{x}_j^\Upsilon, r^L_{f_i:v_i}) \rightarrow \mathbf{x}_j^p + (\mathbf{x}_i^p + r^R_{f_i:\tilde{v}_\mathrm{i}} + \mathbf{x}_i^s) + \mathbf{x}_j^s$$

The new rule is modified to;

$$R_j: (\mathbf{x}_j^\Upsilon, r^L_{f_i:v_i}) \rightarrow \mathbf{x}_j^p + (\mathbf{x}_i^\Upsilon, r^L_{f_i:v_i}) + \mathbf{x}_j^s$$

- Merge or combine rules if they have common feature-value pairs that will construct a more generic rule with respect to their LHSs and RHSs. This will results in reducing the pattern matching time and the complexity of the generated rules.
  For example the following two grammar rules have different LHS but similar (f_i, v) pair at their RHS. The two rules can be resolved by replacing them with a that includes the value of this pair as unbound, e.g. $(f_i, \_)$.

$$R_i\colon \left(\mathbf{x}_i^{\Upsilon}, r_{f_{i}:v_i}^{L}\right) \rightarrow \mathbf{x}^p + r_{f_{i}:v_{\mathrm{i}}}^{R} + \mathbf{x}^s$$

$$\tilde{R}_i\colon \left(x_i^{\Upsilon}, r_{f_{i}:\tilde{v}_i}^{L}\right) \rightarrow \mathbf{x}^p + r_{f_{i}:v_i}^{\tilde{R}} + \mathbf{x}^s$$

The combined rule is

$$R_i\colon \left(\mathbf{x}_i^{\Upsilon}, r_{f_{i}:\_}^{L}\right) \rightarrow \mathbf{x}^p + r_{f_{i}:v_{\mathrm{i}}}^{R} + \mathbf{x}^s$$

Using the rule base for morphological analysis has become quite simple:

- Convert the inflected Arabic word into its vocal representation in an abstract form that consists of consonants and vowels.
- Index each letter in the input word its position.
- Search the rule base for the rule that better matches the vocal representation.
    - If no rule matches found for the vocal representation, the parser looks for the most similar rule using Levenshtein's weighted distance (1966) estimated by Wagner and Fisher algorithm (Wagner & Fisher, 1974).
- Retrieve this rule from the rule base.
- Fire (execute) the rule to build a new word feature-value pairs according to the fired rule and produce the stem.
- Lookup the stem in the lexicon to return the stem and its features.

Example for the analysis process:

Let the tested inflected word be 'يهتدون' (they discover [masculine]).

Convert it to vocal characters so it becomes 'V0C1C2C3V4C5'

Search for the vocal string in the LHS of the generated grammar rules

It matches with the LHS = 'ونC3C2C1ي' , So the matching rule is:

ونC3C2C1ي
[type:verb..tense:present..sex:male..count:plural..person:Third]→ ون + ى C3 C2 C1 إ
[type:verb..tense:past..sex:male..count:singular..person:Third] + ي

by this rule we deduce that the root is defined by removing *yaa* 'ي' character from the first of the word and adding *alef* 'أ' character at the first of the word and *alef layena* 'ى' at the end of the word , so we can search for it in dictionary by word 'إهتدى'

The changed features of the analyzed word according to the rule are the tense from past to present , and the count from singular to plural , and the other features either having constant values or the same as the stem like the person *feature* which is third person for stem and morphed words.

4. Experimental Results

In this section we discuss the results of an experiment that shows the convergent rate of the rule generation algorithm as the sample size increases. The experiment simply runs the proposed leaning algorithm on a sample set of 2500 words separated into several subsets of 500 each, the data is gathered from a raw corpus captured from news site (www.alarabiya.net) and tagged as part of speech (POS) using Arabic tagset described in (Khoja, Garside, Knowles, 2001).

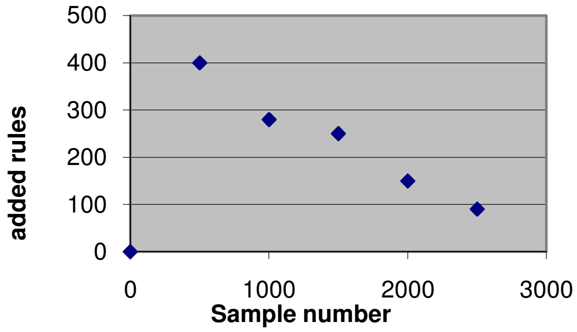


Figure 2: Number of generated rules against the sample size

The learning results of running algorithm that acquires Arabic morphological rules using a set of example pairs that forms a sample set of size 2500 word has produced 1160 morphological rules. They are depicted in Figure 2 using the results presented in Table 1. The results show that the changes in the number of rules as the data set increases.

No. Of Tokens	No. Of Rules
0 – 500	400
500 – 1000	280
1000 – 1500	250
1500 – 2000	140
2000 – 2500	90
Total	1160

Table 1: Number of generated rules against the change in the sample size

To check the effectiveness of the generated grammar, two sets of data are checked against induced rules. First set is the data used to train the system and is called positive data. Second set is raw data collected randomly from a news website.

Analyzing the results of the first set, 98% of the training words correctly matched the induced rules and 2% of them have wrong matching or no matching rule at all. Almost 75% of the training words matched one rule only and 25% matched more than one rule, which leads to ambiguity that should be resolved. Analyzing the results of the second set, 80% of the raw morphed words correctly matched the induced rules and 20% of them have wrong matching or no matching rule at all. Almost 70% of the raw morphed words matched one rule only and 30% of them matched more than one rule, which leads to ambiguity.

5. Conclusions and Future Work

Arabic is morphologically rich language. Automated acquisition of Arabic morphology is a challenge task as Arabic has many peculiarities, which make the traditional manual acquisition problematic. This paper has discussed the development of a learning system for Arabic morphological rules acquisition from Arabic examples. We provided an algorithm which acquires the morphological rules in an efficient representation. We showed how morphological analysis and hence generation could be easily benefit from the repository of the rule base. We conducted an experiment on a sample of 2500 words which shows the learning curve of rules decreases as the number of words increases.

There are many advances to be considered in future to enhance the results of the learning algorithm. These advances include increasing the level of details for the speech vocal classification, integration with statistical based model in a hybrid model and ambiguity resolution.

References

Yona, S., Wintner, S. (2007). A finite-state morphological grammar of Hebrew, *Natural Language Engineering*, 14(2), pp. 173--190.

Mohri, M., Roark, B. (2006). Probabilistic Context-Free Grammar Induction Based on Structural Zeros, In *Proceedings of the Seventh Meeting of the Human Language Technology conference*.

Edelman, S., Solan, Z., Horn, D., Ruppin, E. Rich (2003). Syntax from a raw Corpus: Unsupervised Does it, Presented *at NIPS workshop on Syntax, Semantics and Statistics*.

Pappa, G. L., Freitas, A. (2004). Towards a genetic programming algorithm for automatically evolving rule induction algorithms. In *Proceedings of the Workshop W8 on Advances in Inductive Learning*, pp. 93--108,

Khoja, S., Garside, R., Knowles, G. (2001). A tagset for the morphosyntactic tagging of Arabic, In *Corpus Linguistics Conference*.

Daille, B. (2000). Morphological Rule Induction for Terminology Acquisition, In *Proceedings of the 18th International Conference on Computational Linguistic*.

Cercone, N., An, A., Chan, C. W. (1999). Rule-Induction and Case-Based Reasoning: Hybrid Architectures Appear Advantageous, *IEEE Trans. Knowl. Data Eng.*, 11(1), pp. 166--174.

Silva, R. B., Ludermir, T. B. (1999). Neural Network Methods for Rule Induction, In *Proceedings of International Joint Conference on Neural Networks*.

Mikheev A. (1997). Automatic rule induction for unknown-word guessing, *Computational Linguistics*, 23(3), pp.405--423.

Beesley, K. R. (1996), Arabic Finite-State Morphological Analysis and Generation, In *Proceedings of the 16th Internationa Conference on Computational Linguistics*, pp. 5--9.

Miller, S., Fox, H. J. (1994). Automatic Grammar Acquisition, In *Proceedings of Human Language Technology Conference*.

Uszkoreit, H. (1986). Categorial Unification Grammars, In *Proceedings of International Conference on Computational Linguistics*, pp. 187--194.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, 10(8), pp. 707--710.

Wagner, R. A., Fisher, M. J. (1974). The String-to-String Correction Problem, *Journal of ACM*, 21(1), pp. 168--173.

Muggleton, S. (1999). Inductive Logic Programming: Issues, Results and the Challenge of Learning Language in Logic, *Artificial Intelligence*, 114(1-2), pp. 283--296.