WORKSHOP PROGRAMME

Monday 17 May 2010

9:15-9:30 **Welcome and Introduction**
Khalid Choukri, Owen Rambow, Bente Maegaard, and Ibrahim A. Al-Kharashi

Oral Session 1: Syntax, Semantics, and Parsing

9:30-9:50 **Structures and Procedures in Arabic Language**
André Jaccarini (1), Christian Gaubert (2), Claude Audebert (1),
(1)Maison méditerranéenne des sciences de l'homme (MMSH), France
(2)Institut français d'archéologie orientale du Caire (IFAO), Cairo, Egypt

9:50-10:10 **Developing and Evaluating an Arabic Statistical Parser**
Ibrahim Zaghloul (1) and Ahmed Rafea (2)
(1) Central Lab for Agricultural Expert Systems, Agricultural Research Center, Ministry of Agriculture and Land Reclamation.
(2) Computer Science and Engineering Dept., American University in Cairo

10:10-10:30 **A Dependency Grammar for Amharic**
Michael Gasser
Indiana University, USA

10:30-11:00 **Coffee break**

11:00-12:20 **Poster Session 1: Morphology & NLP Applications I**

A syllable-based approach to Semitic verbal morphology
Lynne Cahill,
University of Brighton, United Kingdom

Using the Yago ontology as a resource for the enrichment of Named Entities in Arabic WordNet
Lahsen Abouenour (1), Karim Bouzoubaa (1) and Paolo Rosso (2)
(1) Mohammadia School of Engineers, Med V University Rabat, Morocco
(2)Natural Language Engineering Lab. - ELiRF, Universidad Politécnica Valencia, Spain

Light Morphology Processing for Amazighe Language
Fadoua Ataa Allah and Siham Boulaknadel
CEISIC, IRCAM, Madinat Al Irfane, Rabat, Morocco

Using Mechanical Turk to Create a Corpus of Arabic Summaries
Mahmoud EL-Haj, Udo Kruschwitz and Chris Fox
School of Computer Science and Electronic Engineering, University of Essex, United Kingdom

DefArabicQA: Arabic Definition Question Answering System
Omar Trigui (1), Lamia Hadrich Belguith (1) and Paolo Rosso (2)
(1) ANLP Research Group- MIRACL Laboratory, University of Sfax, Tunisia
(2) Natural Language Engineering Lab. – EliRF, Universidad Politécnica Valencia, Spain

12:20-13:50 **Lunch break**

13:50-15:10 **Poster Session 2: Morphology & NLP Applications and NLP Tools**

Techniques for Arabic Morphological Detokenization and Orthographic Denormalization
Ahmed El Kholy and Nizar Habash
Center for Computational Learning Systems, Columbia University, USA

Tagging Amazigh with AncoraPipe
Mohamed Outahajala (1), Lahbib Zenkouar (2), Paolo Rosso (3) and Antònia Martí (4)
(1) IRCAM,
(2) Mohammadia School of Engineers, Med V University Rabat, Morocco,
(3) Natural Language Engineering Lab. - ELiRF, Universidad Politécnica Valencia, Spain,
(4) CLiC - Centre de Llenguatge i Computació, Universitat de Barcelona, Barcelona, Spain

Verb Morphology of Hebrew and Maltese - Towards an Open Source Type Theoretical Resource Grammar in GF
Dana Dannélls (1) and John J. Camilleri (2)
(1) Department of Swedish Language, University of Gothenburg, Sweden;
(2) Department of Intelligent Computer Systems, University of Malta, Malta

Syllable Based Transcription of English Words into Perso-Arabic Writing System
Jalal Maleki
Dept. of Computer and Information Science, Linkping University, Sweden

COLABA: Arabic Dialect Annotation and Processing
Mona Diab, Nizar Habash, Owen Rambow, Mohamed Al Tantawy and Yassine Benajiba
Center for Computational Learning Systems, Columbia University, USA

A Linguistic Search Tool for Semitic Languages
Alon Itai
Knowledge Center for Processing Hebrew, Computer Science Department, Technion, Haifa, Israel

13:50-15:10 **Poster Session 3: Speech & Related resources**

Algerian Arabic Speech database Project (ALGASD): Description and Research Applications
Ghania Droua-Hamdani (1), Sid Ahmed Selouani (2) and Malika Boudraa (3)
(1) Speech Processing Laboratory (TAP), CRSTDLA, Algiers, Algeria;
(2) LARIHS Laboratory, University of Moncton, Canada;
(3) Speech Communication Laboratory, USTHB, Algiers, Algeria.

Integrating Annotated Spoken Maltese Data into Corpora of Written Maltese
Alexandra Vella (1,2), Flavia Chetcuti (1), Sarah Grech (1) and Michael Spagnol (3)
(1)University of Malta, Malta
(2)University of Cologne, Germany
(3) University of Konstanz, Germany

A Web Application for Dialectal Arabic Text Annotation
Yassine Benajiba and Mona Diab
Center for Computational Learning Systems, Columbia University, USA

Towards a Psycholinguistic Database for Modern Standard Arabic
Sami Boudelaa and William David Marslen-Wilson
MRC-Cognition & Brain Sciences Unit, Cambridge, United Kingdom

Oral Session 2 : Resources and tools for Machine Translation

15:10-15:30 **Creating Arabic-English Parallel Word-Aligned Treebank Corpora**
Stephen Grimes, Xuansong Li, Ann Bies, Seth Kulick, Xiaoyi Ma and Stephanie Strassel
Linguistic Data Consortium, USA

15:30-15:50 **Using English as a Pivot Language to Enhance Danish-Arabic Statistical Machine Translation**
Mossab Al-Hunaity, Bente Maegaard and Dorte Hansen
Center for Language Technology , University of Copenhagen, Denmark

15:50-16:10 **Using a Hybrid Word Alignment Approach for Automatic Construction and Updating of Arabic to French Lexicons**
Nasredine Semmar
CEA LIST, France

16:10-16:30 **Coffee break**

16:30-17:20 **General Discussion**
Cooperation Roadmap for building a sustainable Human Language Technologies for the Arabic language within and outside the Arabic world.

17:20-17:30 **Concluding remarks and Closing**

Editors & Workshop Chairs

Workshop general chair:

Khalid Choukri, ELRA/ELDA, Paris, France

Workshop co-chairs:

Owen Rambow, Columbia University, New York, USA
Bente Maegaard , University of Copenhagen, Denmark
Ibrahim A. Al-Kharashi, Computer and Electronics Research Institute, King Abdulaziz City for Science and Technology, Saudi Arabia

Table of Contents

Author Index

Structures and Procedures in Arabic Language

André Jaccarini, Christian Gaubert [*], Claude Audebert,
Maison méditerranéenne des sciences de l'homme (MMSH)
5 rue du Château de l'Horloge BP 647 13094 Aix-en-Provence, France
[*] Institut français d'archéologie orientale du Caire (IFAO),
37 el Cheikh Aly Yousef Str., Cairo, Egypt
E-mail: jaccarini@mmsh.univ-aix.fr, cgaubert@ifao.egnet.net, claude.audebert@gmail.com

Abstract

In order to demonstrate the efficiency of the *feedback* method for the construction of finite machines (automata and transductors) applied to the Arabic language, and to exhibit the algebraic characterization of this language through mathematical theory of Schutzenberger school, we have chosen applications which are linked to many domains: morphological analysis (with or *without* lexicon), syntactic analysis, construction of operators for I.R. and noise filtering.
A data bank of finite machines can only be efficient if integrated in a computational environment allowing the extraction of these operators (which are fragments, pieces of operational grammars**)** which are to be combined in order to synthesise new operators, according to the needs.
We have developed a software called *Sarfiyya* for the manipulation of arabic automata.
We constructed an extractor of quotations and reported discourse. The evaluation of this automaton will be available online. *Sarfiyya* was entirely written in Java, which allowed the creation of a Web based application called *Kawâkib*, offering among other functions, root extraction and tool word detection.
We are now heading towards content analysis and text characterization.

1. General presentation[1]

One of the important ideas of our work that Arabic, Semitic languages in general, has a particularly high degree of surfacing "*algorithmicity/grammaticalness"*. We should for this however clarify the relation "procedure/structure" (see below §2). *The structural* characteristics of Arabic language are thus also *algorithmic*. This duality is easily translatable within the framework of the *algebraic theory of the automata* and offers extremely interesting applicative prospects (the correspondence is done in the two directions; see below 2.), easily specifiable.
A certain deficit of mathematical specification is, indeed, one of the characteristics of the actual position of the automatic treatment of Arabic. The theoretical unification, operated thanks to the *algebraic theory of the automata*, seems to us to be particularly interesting to firmly draw up the Arab Studies in the universe of knowledge which is ours today, namely that of the Turing machines. We thus seek to register the automatic treatment of Arabic in the *algebraic* tradition which, in data processing, was especially initiated by M.P Schutzenberger and its school.
In order to support our assumption that the strong algorithmicity/grammaticaness is an important specificity of Arabic we have stated in our preceding studies (Audebert, Jaccarini 1994, Jaccarini 1997, Gaubert 2001) that the construction of parsers, only requiring a minimal recourse to the lexicon, and even in certain cases, completely avoiding the lexicon, did not cause explosions of ambiguities. It is noted indeed that *this passage to the limit* does not present a significant difference with regard to the coefficients of ambiguities of the considered forms compared to the parsers which resort systematically to lexicon (see for instance DIINAR project of the university of Lyon). The context, i.e. syntax is indeed much more interesting on this level, has led us to study in a formal way the passages of information between syntax and morphology.
The minimal recourse to the lexicon even the reduction of all the lexemes to their simple patterns (principle of the empty dictionary) is compensated by the headlight role which we confer on the tokens (tools words) and that we regard as true operators defined by minimal finite machines (automata and transducers). These elements, which are the most constraining, precisely coincide with the fixed elements (the morphological atoms, which do not have roots: for example *inna*, *limādha,* … etc). This *coincidence* has a simple explanation: the cardinality of their class of syntactic congruence[2] is very limited (often equals with the unit) contrary to those of the other lexemes, which can "commutate" with other elements belonging to the same "category" (or class of syntactic congruence) without doubting the "grammaticality" of the sentence, nor its type (as for the relationship between syntactic congruence and the congruence induced by the patterns, refer to "*Algorithmic Approach of Arab grammar"*, first chapter "Morphological System and syntactic monoïd"; to be published; summary of the chapter is available on the theoretical website associated with the article, from now *automatesarabes*). An algebraic characterization of the tokens is given there: they are the "lexical" invariants of the projection of the language on its skeleton (in other

[1] [1] The authors thank Eva Saenz Diez for having read this text. Without her active participation, this work would have been completed.

[2] A syntactic class of congruence consists of all the words that can permute in a sentence without questioning its grammaticality.

words invariants of projection L→ L/RAC). The term "token" was selected in reference so what data-processing originators of language indicate by this term: association of symbols which it is necessary to regard as fixed (example: BEGIN, GOTO, LOOP,…etc), which naturally induce particular "waitings". Thus this approach was initially presented like a "Grammar of waitings" ("grammaire des attentes"; see Audebert, Jaccarini, 1986); this name seems to be perfectly appropriate if one is situated at the level of didactic and of cognition and we thus use it in all the contexts where it is not likely to produce any ambiguity.

The "tokens" can be considered, to some extent, like the "lexemes" of the quotient language. By choosing this term our first concern was simply to draw the attention to the analogy with formal languages. The object L/RAC is a semi-formal language obtained, by projection, starting from a natural language. It presents the particularity to have a very limited lexicon. It is indeed this fact that seems to us the most characteristic of the Semitic system (strong grammaticaliness/algorithmicity) whose Arab language is the current richest representative, best known and especially most spoken.

The assumption of the construction of a syntactic monitor, which the theoretical goal is to study and modeling of the interactions between syntax and morphology and the practical finality –the short-circuiting of the superfluous operations in the location of the structure of the sentence-, remains the long-term objective which will lead the continuation of this research.

Grammars not being static, but being regarded as a particular point of view on the language, they can appear drifting by transformation of a non-fixed core itself. These points of view, i.e. these grammars, can be connected to each other. The question of their adequacy compared to a given objective arises then: the grammar of an orthographical controller is not the same as a program of learning Arabic or of an information extractor.

The morpho-syntactic analysis of Arabic which we propose (see bibliography) constitutes a reflection and a general tool to answer a whole set of applications.

The development of this approach passes by a thorough study of the tokens or words tools. This study results in the constitution of morphosyntaxic grammars, conceived like *operational* bricks of grammars *in order to synthesize procedures of research*.

Such grammars built from automata and finite transducers can also be used to detect various types of sentences, for example conditional sentences, relations of causality or other discursive relations, to extract the reported speeches, etc. These topics are fundamental for the extraction of information: coupled with the search for collocations (non-fortuitous co-occurrences), in interaction with the morpho-syntactic analyzer, they are the source of many applications. We presented to MEDAR 09 (Audebert, Gaubert, Jaccarini, 2009) examples of constructions of such operational grammars that is at the level of the morphological, syntactic analysis or even at the level of extraction of information (I.R). As simple illustrative example, we have introduced an operator of extraction of speeches, built from finished states machines (automata and transducers). This machine that we made deterministic by a calculation, requiring a time of several hours but carried out only once and whose result is then stored in memory once for all (the final automaton contains nearly 10.000 states!) allows to extract in a reasonable period time (which will be improved later on) all the quotations from a text of almost 30.000 words with a rate of success of almost 80% if one makes however abstraction of silences and noises especially due to anomalies of punctuation and a lack of standardization to which it will be more or less possible in the future to mitigate. Rate of success and times show that the operation of feasibility was a success (see herewith table reproduced on *automatesarabes*). An evaluation is thus provided, which proves that our formal considerations, even algebraic, are indeed necessary to make coherent the theoretical framework, without which one is likely to be involved in a pragmatism which ends up becoming sterile. At the present time, we are very much conscious of the disadvantages of *ad hoc* programming.

2. Advantages of the representation of Arabic by finite machines

2.1 The transfer from structures to procedures (Arabic language <--> automata)

This transfer is carried out in the two directions. It is possible indeed, on the theoretical level to establish a formal link between the structural and *algebraic* properties by which we characterized the Arab system, on one hand, and automata and transducers on which we based our algorithmic approach of Arabic, on the other hand. These automata and transducers constitute a class of machines equivalent to those of Turing, on which we nevertheless introduced *restrictions* and conditions of *minimality* related to a rigorous hierarchy.

The Arabic structural properties that we proposed relate to the commutation of the operators of "categorization" and "projection". "Categorization" amounts building a quotient unit made up of the "classes of syntactic congruence" on the free monoïd of the symbols of the initial vocabulary (the formal words in the case of syntax), or "syntactic categories". The relation of implied congruence formalizes the *distributional* principle of linguists (Bloomfield): it expresses that two words can be regarded as equivalents if and only if they can commutate without affecting the grammaticalness of the sentence. Projection amounts reducing the word to its pattern, which also induces a partition in class of congruence: The concatenations (reciprocally the segmentations) remain *invariant* even by changing the roots. This last congruence is compatible with the preceding one in the sense that the syntactic relation of congruence is more "coarse" (the least fine) that can be defined on the monoïd made up from Arabic graphemes and who saturates the unit consisted by the unit or Arabic graphic words, which has as a corollary that any patter, considered here as a given class of words and not as an

operator generating them (duality structure-procedure), is necessarily located inside a syntactic category within the framework of the unit of the Arab graphic words. We registered the modeling of the Arab morph-syntax within the general framework of a *principle of invariance* deriving from the previous Arabic morphographic property, which is obvious (invariance by change of root within the framework of the morphography) by generalizing it with syntax, namely that:
1. To establish syntactic classes which are partitioned in patterns
2. or, on the contrary, to establish patterns which can then be gathered into syntactic classes.
It is the same.

Let us suppose that Π and SC respectively indicate canonical homomorphisms associated with congruences considered higher: syntactic congruence and the one associated with the patterns (categorization and projection), then the principle of invariance will be able to be expressed in an even more concise way: that of the commutation of these two last "operators":

$$\Pi.SC = SC.\Pi$$

Following this principle, we will thus categorize so that the construction of the grammar is not affected by the operation which consists in reducing the language only to its paradigms (patterns + tokens).
The possibility of building a *computer program functioning without lexicon* is only one *consequence* of the above-mentioned property according to which it *should* be indifferent to first categorize and then project or vice versa.

In addition the definition of the automata, as those of the machines of Turing, can appear somewhat *contingent* - but it is quite astonishing that such a rough mechanism can represent all calculations that can be made on machine and even that it can (theorem of the universal enumeration) simulate any computer or set of computers (including Internet!). The automata with two stacks of memories (we don't need a large number of them, which does represent a remarkable property) are equivalent to these machines. These automata are founded on those of less complexity, without stacks of memory: the finite-state machines whose definition can cause the same feeling of "uneasiness" mentioned above –while talking about the machines of Turing- and at the same time amazement due to the fact that such an elementary mechanism can generate such complex configurations.

The adaptation of a *more abstract* or algebraic viewpoint, allows us at the same time
1. to avoid this uneasiness of contingency and
2. to give a meaning to the extension of the principle of invariance from the linguistic level to the data-processing level, to thus unify the theoretical framework while offering extremely interesting practical prospects. Indeed the calculation of the monoïd of transition M(L) from the language L means building the minimal deterministic automat *directly* accepting this language. One will find in the *automatesarabes* website, the development on an example of syntax with this type of calculation (taken from "Linguistic Modeling and Automata Theory", see *automatesarabes*). This illustration offers a theoretical interest (to reduce a possibly infinite set of sentences to a finished number of configurations) as well as a practical one (the "automatic" construction of the minimal deterministic automat corresponding).
The automaton corresponding to the study of David Cohen (Cohen, 1970) will be rebuilt by using this same method (which leads to the constitution of an automat of 13 states and 102 transitions) while following an "entirely automated chain" if we may say so, or rather "automatisable".

Any sequence of a language can indeed be regarded as an application of an initial segment of N in itself and to say that a language is recognizable by a finite-state automaton it is in fact equivalent to define a *congruence* on this language whose set of classes is finite

The theorems which explicitly establish the links between the concepts of syntactic monoïd, congruence and the traditional concept of automaton, such as we use them for our analysis of Arabic, also appears in *automatesarabes*.

In conclusion the syntactic monoïd, with which is associated a minimal deterministic automaton being able to recognize this language can be produced thanks to a transducer[3]. This monoïd of transition (= syntactic) can be obtained automatically.

2.2 Automatic vocalisation and transduction

This second point deserves to be insulated, given its importance. The standard writing of Arabic is shorthand. The short vowels are not noted, which has as a natural consequence to increase considerably the ambiguity and the difficulties of reading. Moreover cases are often marked by short vowels, if they are singular, and their "calculation" are not always extremely easy[4].

[3] We had programmed it in Lisp; the question of its restoration is posed today in terms of opportunity, utility, calendar, working time, etc, etc. For the moment this task is not a priority. It is also possible to enhance this transducer (minimal) in order to determine the basic relations, which associated with the generators, define the monoïd of transition (isomorphous with the syntactic monoïd). It can indeed be interesting to have the possibility of defining an infinite language (determined nominal group or not determined, Conditionals, etc.) by a small set of limited equalities relating to this language of limited length, rather than by rewriting rules. For example in the example evoked in the site (a small subset of the given nominal group), in order to check that an unspecified sequence belongs to this language, it is enough to examine its sub-sequences of length 3.

[4] The great grammarian Sībawayh quotes an *irreducible* example of ambiguity, very much like a joke, which has also the merit to draw the attention to the fact that in literary Arabic the *place* of the words in the sentence is relatively free; this freedom being compensated by a more important morpho-casual "marking". The example

Our matter is not to discuss the relevance of this written form but to note the phenomenon while trying to measure its consequences in term of ambiguities and to provide *objective* arguments for the supporters of the two camps: the one constituted by the "Arabists" whose systems of transliteration, which they use in general, do not leave any right to the error (short vowels having the same value as the full consonants or the long vowels (there are only three)) and the usual system used by Arabs that leaves a certain latitude[5] which seems to suggest - a fact corroborated by the outside experiments (but which remains to be more deeply examined) - that the reading (without diacritics) in the first flow makes it possible to perceive the meaning of the sentence overall; a finer syntactic analysis, implying backtracking allows in the second time to raise ambiguity.

Nevertheless these assumptions must be evaluated. The system of transduction based on underlying automata, to which we can make correspond "semantic attributes" of grammars of Knuth grammars (see *automatesarabes*), or "schemes guided by syntax" (Aho, Seti and Ullman, 1986), which are associated with synthesized" and "inherited" attributes, is particularly well adapted for this task (linear dominating flow "blocked", nevertheless, by "backtracking" which can present in the most dramatic cases, vicious circles (deadlock) i.e impossibilities of vocalization (irreducible ambiguities which are cases to be studied for itself)). The synthesized attributes are values that are propagated from bottom to top of the tree representing the structure of the sentence (it is said that one decorates the tree or even that one associates to himself a "semantic" tree) and the inherited attributes, those which are propagated from top, downwards. Transposed to the reading flow, that means, there exist values (here one is interested in the short vowels) which "are synthesized" progressively according to the advancement of the reading head, whereas certain lexeme can only acquire their final value (vocalization or meaning, …) by the retroactive effect, once the complete reading of the sentence was accomplished. Knuth studied the cases of vicious circles and developed at the item (1968) an algorithm to avoid them. In the case of impossibility, you then find yourself in the well-known case of the data processing specialist, the "deadlock", which occurs when two processes are on standby, one of the other. It is an *intrinsic* ambiguity[6].

In "Algorithmic Approach of the Arabic Grammar" (see *automatesarabes*), we have presented an *ambiguous* morphological transducer, functioning word by word (vowels dependant on the case (linguistic) are not being taken into account, since the connection with syntax was not implemented[7]). Coefficients of ambiguity are varying from 1 (in a significant number of cases[8]) up to 12.

It is obvious that a connection with syntax is necessary not only to cause a drop in the level of ambiguity but also to be able to vocalize the end of the words.

Such tools that are to be reprogrammed can already have extremely interesting applications. The writer of an Arabic text can be informed in real-time of the level of ambiguity of the form introduced to see himself suggested a certain number of solutions (total or partial) to reduce ambiguity according to the *level of user* (tutorial), only by clicking. Fundamental technology already exists; all the rest is only question of ergonomics and interface, which in this field is *fundamental*.

It goes without saying that it would be an improvable tool and evolvable tool by introduction of syntax but also by training.

The conceptual tool (the interactive transducer of vocalisation) would obviously be of greater interest to answer the question that had been asked at the beginning of this paragraph namely to try to measure or rather to scientifically discuss the relevance of the two viewpoints: respectively the Arabists one and Arabs conception, to say it in a concise way.

It would have been difficult to scientifically discuss this question of relevance if one had not had recourse to the transducers functioning letter with letter and interacting with the highest level "director" automats: the syntactic automats.

2.3 Transparency of the analyzers

The transparency of the analyzers which can be entirely specified mathematically, offers essential advantages that we will only mention here: those to offer evidence of programs as well as measurements of complexity and, last but not least, the possibility of establishing relevant similarities with the natural process of apprehension

is the following: Akala (ate) IssA (Jesus) MoussA (Moses); one cannot know if it is Jesus who ate Moses or the reverse, being given the phonological incompatibility of the mark of the direct case (short vowel *u*) with the last phoneme of IssA or MoussA. The ambiguity naturally remains the same one by permutation of Issa and of MoussA (the mark of indirect being short *a*).

[5] This report of the use of standard writing by the Arabs since centuries, as well as their organization of their dictionaries, makes us naturally think that they perceived (and continue to perceive) consonants as being elements of a "skeleton" which would be the principal "*support*" of the meaning (more specifically the radical consonants, the others being able to belong to a pattern, inevitably discontinuous, if we take into account the short vowels which inevitably intervene there, which can never be radical; the pattern in its *entirety*, which is a non-concatenative form, being only (with the root) likely to have one or more, semantic values). In this remark we are within the framework of morphology known as *healthy*. We announce only facts and we voluntarily keep away from the problem of lexical "solidification" (fixation).

[6] This question also arises about the "syntactic monitor" which is supposed to optimize the morphological analysis, where we must consider the extreme case where both morphological and syntactic processors are waiting for each other (irreducible ambiguity).

[7] Some results will be available on the site *automatesarabes* before the publication of the book.

[8] However no statistics were drawn up; it was about a feasibility study.

(cognitivo-algoritmic parallelism).

3. Coming back to tokens: from syntax to semantics

Study of the syntactic operators and grammar of syntactic waitings.

1. The pivot of this research is the study, made easier by an advanced version of the tool *Sarfiyya*, of the grammatical waitings of all the tokens (word tools of Arabic), whose location is already solved. For example, the operator *inna* implies the presence of structures having precise grammatical functions (topic, predicate) that are recognizable by the machine. On the other hand, prepositions (*ʿalā*, *ʿinda*) are of more reduced range but can possibly combine with high level tokens: a hierarchy is established between families of operators. It is necessary *to formalize the* syntactic behaviors and their *local and total* implications.
This research was started on a corpus and remains to be undertaken. It is essential for the study of the syntax of Arabic and, although outlined it has to be reset once again. The number of the tokens amounts to approximately 300 and poses problems of location dealt according to a certain methodology and raises, by definition, questions concerned with syntax whose modeling must taken into account.
2. This study will be coupled with that of the linguistic markers of certain **discursive relations**. This work consists in creating a base of the most possible elementary automats (or transducers), so that their combinations can allow the *synthesis* of new functionalities of search for information (IH). A first demonstration of the effectiveness of this method was provided (MEDAR 09). The *progressive* refinement of the filters and the reduction of the noises were obtained, according to a precise experimental method, consisting in retroacting to the initial grammar according to the result provided by the machine. This method of *feedback* (continual coming and going between theoretical modeling and implementation) naturally supposes a work **of evaluation of grammars**.
However, there exists several manner of assigning a value to a grammar, according to the standard selected, which varies according to the required application. The standard allows assigning to the grammar a **value** starting from fixed criteria. A criterion can be essential for a given application but not very relevant for another (for example the non-ambiguous extraction of the root represents only little interest if the objective is to obtain a simple spellchecking). The data of the standard makes it possible to privilege, according to its needs, certain criteria among others and thus induces a hierarchy.
Inheriting its code from *Sarfiyya* with some enhancements for collaborative work, the web-based application *Kawâkib* and its latest version *Kawâkib Pro* (fig. 1) are the tools we use for now to collect linguistic data connected with tool words, to parse pieces of corpus with automata and to perform measures in this regard. It also includes tools for root searches, frequencies reports, etc.

We will find in *automatesarabes* a detailed evaluation of the extractor of quotations in journalistic texts (which is extremely encouraging). This experiment constitutes a starter of the pump of feedback announced in MEDAR 09.

4. References

Automatesarabes: http://automatesarabes.net.

Aho, Sethi, Ullmann (1986), Compilateurs. Principes, techniques et outils. French edition 1991. InterEdition.

Audebert C, Jaccarini A. (1986) À la recherche du Ḫa*bar, outils en vue de* l'établissement d'un programme d'enseignement assisté par ordinateur, Annales islamologiques 22, Institut français d'archéologie orientale du Caire.

Audebert C, Jaccarini A. (1988). De la reconnaissance des mots outils et des tokens. Annales islamologiques 24, Institut français d'archéologie orientale du Caire.

Audebert C, Jaccarini A. (1994). Méthode de variation de la grammaire et algorithme morphologique. Bulletin d'études orientales XLVI. Damascus.

Audebert, Gaubert, Jaccarini (2009). Minimal Ressources for Arabic Parsing/ an Interactive Method for the Construction of Evolutive Automata. MEDAR 09.(http://www.elda.org/medar-conference/summaries/37.html)

Audebert (2010). Quelques rélexions sur la fréquence et la distribution des mots outils ou tokens dans les textes arabes en vue de leur caractérisation dans le cadre de l'extraction d'information. Annales islamologiques 43, Institut français d'archéologie orientale du Caire.

Beesley, Kenneth R. (1996). Arabic Finite-State Morphological Analysis And Generation. COLING.

Cohen, D. (1970) Essai d'une analyse automatique de l'arabe. In: David Cohen. Etudes de linguistique sémitique et arabe. Paris:Mouton, pp. 49-78.

Gaubert Chr., (2001). Stratégies et règles pour un traitement automatique minimal de l'arabe. Thèse de doctorat. Département d'arabe, Université d'Aix-en Provence.

Gaubert (2010), *Kawâkib*, une application web pour le traitement automatique de textes arabes, Annales islamologiques 43, Institut français d'archéologie orientale du Caire.

Jaccarini A., (1997). Grammaires modulaires de l'arabe. Thèse de doctorat. Université de Paris-Sorbonne.

Jaccarini (2010). De l'intérêt de représenter la grammaire de l'arabe sous la forme d'une structure de machines finies, Annales Islamologiques 43, Institut français d'archéologie orientale du Caire.

Koskenniemi K. (1983). Two-level Morphology. A General Computational Model for Word-Form Recognition and Production. Department of General Linguistics. University of Helsinki.

fig. 1 : The *Kawâkib Pro* web-based application

Developing and Evaluating an Arabic Statistical Parser

Ibrahim Zaghloul
Central Lab for Agricultural Expert Systems
Agricultural Research Center
Ministry of Agriculture and Land Reclamation.
ibrahimz@claes.sci.eg

Ahmed Rafea
Computer Science and Engineering Dept.
American University in Cairo
rafea@aucegypt.edu

Abstract

This paper describes the development of an Arabic statistical parser using Arabic Treebank and a statistical parsing engine. The different steps followed to develop and test the parser have been described. We divided the LDC2005T20 Arabic Treebank into training and testing sets. 90 % of the treebank was used to train the Bikel parser package while 10% of it was randomly selected to test the developed parser. The testing data set annotations were removed to convert it into pure text to be introduced to the trained parser. The gold testing data set was prepared, by mapping its tags, to the tags produced by the trained parser. This mapping was necessary to evaluate the parser results using a standard evaluation tool. The metrics widely applied for parsers evaluation were computed for the developed parser results. The F-measure evaluation metric of the developed parser was 83.66 % which is comparable to evaluation metrics results of well known English parsers.

1. Introduction

In this paper we present the steps followed to develop and evaluate an Arabic parser using the Dan Bikel multilingual parsing package [1] and the LDC2005T20 Arabic Treebank. The results of testing the parser are presented for sentences with different lengths.

Parsing is the task of identifying one or more tree structures for a given sequence of words (Bikel, 2004). Instead of rule-based parsers, which used hand-crafted grammars, statistical parsers increased accuracy and tend to exhibit greater robustness in dealing with unusual utterances, which would cause a more strictly rule-based parser to fail. They also have the advantage of being easier to build and to customize (Venable, 2003).

Treebank statistical parsers induce their grammar and probabilities from a hand parsed corpus (Treebank). If it is required to have a parser that produces trees in the Treebank style to all sentences thrown at it, then parsers induced from Treebank data are currently the best (Charniak, 1997).

Creating the Treebank is a staggering task, and there are not many to choose from. Thus the variety of parsers generated by such systems is limited. At the same time, one of the positive effects of creating Treebanks is that several systems now exist to induce parsers from this data and it is possible to make detailed comparisons of these systems (Charniak, 1997). Also, the availability of large, syntactically bracketed corpora such as the Penn Tree Bank afforded opportunity to automatically build or train broad coverage grammars (Sekine and Grishman, 1995).

Statistical parsers work by assigning probabilities to possible parses of a sentence, locating the most probable parse, and then presenting that parse as the answer (Charniak, 1997). The probability of each candidate tree is calculated as a product of terms, each term is corresponding to some sub-tree within the tree (Collins, 1999).

In general, to construct a statistical parser one must figure out how to:

a. Train the parser to construct the grammar rules and their probabilities.
b. Find possible parses for new sentences.
c. Assign probabilities to these new sentences.
d. Pull out the most probable parse for each sentence (Charniak, 1997).

Applications that potentially benefit from syntactic parsing include corpus analysis, question answering, natural-language command execution, rule-based automatic translation, and summarization (Venable, 2003).

In our work we used the Dan Bikel multilingual parsing engine. Bikel parser is the only parsing engine, we found, that considers Arabic. It contained some customizations of the general features, which he called 'Language Package', to fit with the Arabic language.

The motivation behind this work was the need of having an Arabic parser in many applications like machine translation, text summarization, and others.

The objective of the work presented in this paper was developing a statistical Arabic parser using a Treebank and a parsing engine, and evaluating the performance of the developed parser.

Section 2, reviews related work in the statistical parsing area. In section 3, Arabic parser development steps are described. In section 4, the evaluation methodology is explained. In section 5, the results of parser testing are shown and discussed.

[1] http://www.cis.upenn.edu/software.html#stat-parser

2. Related work

A lot of work has been done in the statistical parsing area. Most of the work concentrated on parsing English as the main language and paying no or little attention to other languages. The following subsections summarize statistical parsers developed for English.

2.1 Apple Pie Parser

Apple Pie (Sekine and Grishman, 1995) extracts a grammar from Penn Treebank (PTB) v.2. The rules extracted from the PTB have S or NP on the left-hand side and a flat structure on the right-hand side. The parser is a chart parser. The parser model is simple, but it can't handle sentences over 40 words. This parser gave 43.71% Labeled Precision, 44.29% Labeled Recall, and 90.26% Tagging accuracy, when tested on section 23 of the Wall Street journal (WSJ) Treebank.

2.2 Charniak's Parser

Charniak presents a parser based on probabilities gathered from the WSJ part of the PTB (Charniak, 1997). It extracts the grammar and probabilities and with a standard context-free chart-parsing mechanism generates a set of possible parses for each sentence retaining the one with the highest probability. The probabilities of an entire tree are computed bottom-up.

In (Charniak, 2000), he proposed a generative model based on a Markov-grammar (Charniak, 2000). It uses a standard bottom-up, best-first probabilistic parser to first generate possible parses before ranking them with a probabilistic model. This parser gave 84.35% Labeled Precision, 88.28% Labeled Recall, and 92.58% Tagging accuracy, when tested on section 23 of the WSJ Treebank.

2.3 Collins's Parser

Collins's statistical parser (Collins, 1996) (Collins, 1997) is based on the probabilities between head-words in parse trees. Collins defines a mapping from parse trees to sets of dependencies, on which he defines his statistical model. A set of rules defines a head-child for each node in the tree. The parser is a CYK- style dynamic programming chart parser. This parser gave 84.97% Labeled Precision, 87.3% Labeled Recall, and 93.24% Tagging accuracy, when tested on section 23 of the WSJ Treebank.

2.4 Bikel Parser

Bikel based his parser on Collins model 2 (Collins, 1999) with some additional improvements and features in the parsing engine like: layers of abstraction and encapsulation for quickly extending the engine to different languages and/or Treebank annotation styles, "plug-'n'-play" probability structures, flexible constrained parsing facility, and multithreaded for use in a multiprocessor and/or multihost environment.

2.5 Stanford Parser

The Stanford Parser is an un-lexicalized (does not use lexical information) parser which rivals state-of-the-art lexicalized ones (Klein and Manning, 2003). It uses a context-free grammar with state splits. The parsing algorithm is simpler, the grammar is smaller. It uses a CKY chart parser which exhaustively generates all possible parses for a sentence before it selects the highest probability tree. This parser gave 84.41% Labeled Precision, 87% Labeled Recall, and 95.05% Tagging accuracy, when tested on section 23 of the WSJ Treebank (Hempelmann et.al, 2005).

3. Arabic Statistical Parser Development

This section describes the steps for generating the Arabic probabilistic grammar from an Arabic tree bank. The first subsection describes the used Treebank while the second subsection shows how we divide this Treebank into training and testing parts. The third subsection describe the generation of the probabilistic grammar.

3.1 Arabic Treebank

The Arabic Treebank we used is LDC2005T20. The Treebank contains 12653 parsed Arabic sentences distributed among 600 text files representing 600 stories from the An Nahar News Agency. This corpus is also referred to as ANNAHAR. The sentences lengths distributions in the Treebank are shown in Table (1).

Length	Number of sentences
From 1 To 20	4046
From 21 To 30	2541
From 31 To 40	2121
From 41 To 50	1481
From 51 To 60	942
From 61 To 100	1257
From 100 To max	265

Table (1): Sentences lengths (in words) distributions in the Arabic Tree

3.2 Division of Treebank

The gold standard testing set size was selected to be 10% of the Treebank size which is approximately 1200 sentences and the remaining sentences were left for training. The complete description of the selection of the gold standard set is as follows:

- We first grouped all the Treebank files in one file containing all sentences,
- Then, we used a methodology to avoid being biased in the test sentences selection. The methodology was to select a sentence from every 10 sentences; that is we span the Treebank and pick a sentence after counting 9 sentences. This means that the sample we selected is distributed over all the Treebank.
- The selected sentences are put in a separate gold file and all unselected sentences are put in a separate training file. After completing this step we will have two files: the gold data set file and the training data set file.

3.3 Parser Training

The training data set which is approximately 11400 sentences is introduced to Bikel parsing package to generate the Arabic probabilistic grammar. This grammar is used by the parser included in the parsing package to generate the parsing tree for an input sentence.

4. Evaluation Methodology

The Arabic statistical parser will be evaluated following these steps:

1. Select the evaluation tool.
2. Extract the test data set (remove annotations to be pure text) from the gold data.
3. Prepare the test data for parsing.
4. Run parser on the extracted test data.
5. Pre-process the gold data set to meet the requirements of the evaluation tool.

The following subsections describe in some details each of the above mentioned steps.

4.1 Evaluation Tool and Output Description

The evaluation tool "Evalb"[2], was used in evaluating the parser output. It was written by Michael John Collins (University of Pennsylvania) to report the values of the evaluation metrics for a given parsed data.

The description of the outputs of Evalb is as follows:
1) *Number of sentence:* The total number of sentences in the test set.
2) *Number of valid sentences:* Number of sentences that are successfully parsed.

3) *Bracketing Recall:*

$$\frac{\text{Number of Correct Constituents}}{\text{Number of Constituents in the Gold File}}$$

4) *Bracketing Precision:*

$$\frac{\text{Number of Correct Constituents}}{\text{Number of Constituents in the Parsed File}}$$

5) *Bracketing FMeasure*: The harmonic mean of Precision and Recall. FMeasure =

$$\frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

6) *Complete Match:* Percentage of sentences where recall and precision are both 100%.

7) *Average Crossing:*

$$\frac{\text{Number of constituents crossing a gold file constituent}}{\text{Number of sentences}}$$

[2] http://nlp.cs.nyu.edu/evalb/

8) *No Crossing:* Percentage of sentences which have zero crossing brackets.
9)*2 or less crossing*: Percentage of sentences which have two or less crossing brackets.
10) *Tagging Accuracy:* Percentage of correct POS tags.

4.2 Extracting the test data

A tool was developed to extract the test sentences from the gold standard set. This tool takes the gold data file and extracts the words only. So the output is a file containing the sentences words without any annotations, which will then be given to the parser. Each sentence is processed separately by reading the tokens and extracting the word from each token, ignoring any additional annotations or characters.

4.3 Preparing the Test Data for Parsing

The test sentences have to be put in a suitable form for parsing. The Bikel parser accepts the input sentence in one of two formats:

1. (word1 word2 word3 ……. wordn).
2. (word1(pos1) word2(pos2) word3(pos3)… wordn(posn)).

We put all the test file sentences in the format that allows the parser to do its own part of speech tagging, which is the first format.

4.4 Running the Parser

The parser has been run over the 1200 test sentences using the training outputs and the parameters file for Arabic parser.
The parameter "pruneFactor", described below, was set to value 2 instead of the default value 4 in order to increase the parsing speed. This change in parameter value was made because the default value didn't work well for Arabic giving infinite time for long sentences.
The total parsing time for the test set was about 25 minutes on a machine of processor 3GHz and 8 GB RAM.

pruneFactor: Is a property in the parameter file by which the parser should prune away chart entries which have low probability. The smaller the pruneFactor value, the faster the parsing.

4.5 Processing the Gold Data

The gold standard set is processed to be in the evaluation used by the evaluation tool. The reason for this processing is that the Arabic Treebank annotation style was found to be different from the parser annotation style. In the Treebank we had, the part of speech tags used are the morphological Arabic tags. But in the Bikel parser output the tags are from the original Penn Treebank tag set.

The following example shows the sentence:
" fy AlsyAsp , AlAHtmAlAt kvyrp w AlHqA}q mEqdp."
(In politics, there are many possibilities and the facts are complex.)
As represented in the LDC Treebank:

```
(S (S (PP (PREP fy) (NP
(DET+NOUN+NSUFF_FEM_SG+CASE_DEF_GEN AlsyAsp)))
(PUNC ,) (NP-SBJ
(DET+NOUN+NSUFF_FEM_PL+CASE_DEF_NOM
AlAHtmAlAt)) (ADJP-PRD
(ADJ+NSUFF_FEM_SG+CASE_INDEF_NOM kvyrp))) (CONJ w)
(S (NP-SBJ (DET+NOUN+CASE_DEF_NOM AlHqA}q))
(ADJP-PRD (ADJ+NSUFF_FEM_SG+CASE_INDEF_NOM
mEqdp))) (PUNC .))
```

When this sentence is parsed using Bikel parser, the following annotated sentence is produced:

```
(S (S (PP (IN fy) (NP (NN AlsyAsp))) (, ,) (NP (NNS
AlAHtmAlAt)) (ADJP (JJ kvyrp))) (CC w) (S (NP (NN
AlHqA}q)) (ADJP (JJ mEqdp))) (PUNC .))
```

In the Bikel parser training phase, the LDC tags are converted into the Bikel tags using the "training-metadata.lisp" file. Unfortunately this conversion is part of the grammar generation code in Bikel package. Consequently we have to develop a separate program that converts LDC tags into Bikel tags in order to test the parser. The output of this process is the gold file that enables evaluating the output of Bikel parser running on the test data against this gold file.

5. Results and Analysis

We applied the evaluation tool on the whole test set with no length restriction to test the overall quality, and then we made the evaluation again to see the change in the metrics values up or down for different sentences lengths. We examined the results for the parser outputs trying to analyze the reasons for the drop in the accuracy for some metrics for different sentences lengths.

5.1 Results

Applying the evaluation tool on the whole test set with no length restriction, produces the following results:

Metric	Value
Number of sentence	1200
Number of Valid sentence	1200
Bracketing Recall	82.74
Bracketing Precision	84.60
Bracketing FMeasure	83.66
Complete match	18.92
Average crossing	2.92
No crossing	44.67
2 or less crossing	65.58
Tagging accuracy	99.11

Table (2): Evalb output for the whole test set.

We here show the change in the metrics values up or down for different sentences lengths. The results for 100, 60, 40 and 10 words length sentences are shown in table (3).

Metric	<=100	<=60	<=40	<=10
Number of sentence	1180	1089	888	197
Bracketing Recall	83.24	83.49	**83.80**	81.29
Bracketing Precision	85.07	85.15	**85.44**	77.31
Bracketing FMeasure	84.14	84.31	**84.61**	79.25
Complete match	19.24	20.75	25.00	**45.69**
Average crossing	2.63	2.20	1.63	**0.20**
No crossing	45.42	48.58	55.29	**87.82**
Tagging accuracy	**99.10**	99.02	98.81	97.25

Table (3): evalb outputs for the different lengths

5.1 Analysis of the Results

The best accuracy of the parser appears with sentences in the "less than forty" category, as it has the highest F-measure value.
Some metrics values drop at the "less than ten" category like Recall, Precession, F-measure and tagging accuracy. But the Complete match and No crossing metrics go up for this category.

These values went down as sentences less than ten are more sensitive to any error, i.e. the accuracy for a sentence with length 5 words will be 80% accuracy with one wrong bracket or tag, although accuracy will be 87.5% for a sentence with 40 words and 5 wrong brackets or tags.
On the other hand, the chance to have a complete match increases for shorter sentences because it has smaller number of brackets.

6. Conclusion

The results we got show that the Arabic parser we built here gives results comparable to the results obtained for English. The best Labeled Precision of an English parser was 84.97 % obtained by Collins parser while the labeled precision using Bikel parser adapted to Arabic was 84.6 %. The best labeled recall of an English parser was 88.28% obtained by Charniak while the labeled recall using Bikel parser adapted to Arabic was 82.74%. The best tagging accuracy of an English parser was 95.05% obtained by Stanford parser while the tagging accuracy using Bikel parser adapted to Arabic was 99.11 %. It should be noticed that all English results were obtained using sections 02-21 of the WSJ part of the English Treebank for training (39,832 sentences) and section 23 of WSJ for testing (2416 sentences). In our case we run the experiment on 1200 sentences only.

7. References

Brian Roark and Richard Sproat. 2006. Computational Approaches to Morphology and Syntax, Oxford University Press.

Charniak, Eugene. 2000. A maximum entropy–inspired parser. In Proceedings of the 1st NAACL, pages 132–139, Seattle, Washington, April 29 to May 4.

Charniak, Eugene. 1996. Tree-bank grammars, Technical report, Department of Computer Science, Brown University.

Charniak, Eugene. 1997. Statistical parsing with a con text-free grammar and word statistics, In Proceedings of the 14th National Conference on Artificial Intelligence, pages 598–603, Providence, RI.

Charniak, Eugene. 1996. Tree-bank grammars, Technical Report CS-96-02, Department of Computer Science, Brown University.

Charniak, Eugene. 1997. Statistical Techniques for natural language parsing, AI Magazine 18 4 (1997), 33-43.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistic, Sapporo, Japan, Pages 423-430.

Daniel M. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine, In Proceedings of HLT2002, San Diego, CA.

Daniel M. Bikel. 2004. On the parameter space of generative lexicalized statistical parsing models, Ph.D. thesis, University of Pennsylvania.

Hempelmann, Christian F. and Rus, Vasile and Graesser, Arthur C. and McNamara, Danielle S. 2005. Evaluating State-of-the-Art Treebank-style Parsers for Coh-Metrix and Other Learning Technology Environments, In Proceedings of the Second ACL Workshop on Building Educational Applications Using NLP 2005, Ann Arbor, Michigan, Pages 69-76.

Michael John Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing, Ph.D. thesis, University of Pennsylvania.

Michael John Collins, 1997, Three generative lexicalized models for statistical parsing, In Proceedings of the 35th Annual Meeting of the ACL. 1997, 16-23.

Peter Venable. 2003. Modeling Syntax for Parsing and Translation, Ph.D. thesis, Carnegie Mellon University.

S.Sekine and R. Grishman. 1995. A corpus-based probabilistic grammar with only two non-terminals, In proceedings of the International Workshop on Parsing Technologies.

A Dependency Grammar for Amharic

Michael Gasser

School of Informatics and Computing
Indiana University, Bloomington, Indiana USA
gasser@cs.indiana.edu

Abstract

There has been little work on computational grammars for Amharic or other Ethio-Semitic languages and their use for parsing and generation. This paper introduces a grammar for a fragment of Amharic within the Extensible Dependency Grammar (XDG) framework of Debusmann. A language such as Amharic presents special challenges for the design of a dependency grammar because of the complex morphology and agreement constraints. The paper describes how a morphological analyzer for the language can be integrated into the grammar, introduces empty nodes as a solution to the problem of null subjects and objects, and extends the agreement principle of XDG in several ways to handle verb agreement with objects as well as subjects and the constraints governing relative clause verbs. It is shown that XDG's multiple dimensions lend themselves to a new approach to relative clauses in the language. The introduced extensions to XDG are also applicable to other Ethio-Semitic languages.

1. Introduction

Within the Semitic family, a number of languages remain relatively under-resourced, including the second most spoken language in the family, Amharic. Among other gaps in the available resources, there is no computational grammar for even a sizable fragment of the language; consequently analysis of Amharic texts rarely goes beyond morphological analysis, stemming, or part-of-speech tagging.

This paper describes a dependency grammar for a fragment of Amharic syntax. The grammar is based on Extensible Dependency Grammar (XDG), developed by Ralph Debusmann and colleagues (Debusmann et al., 2004; Debusmann, 2007). XDG was selected because of its modular structure, its extensibility, and its simple, declarative format. The paper begins with an overview of XDG and a description of some relative aspects of Amharic morphosyntax. Then we look at the extensions to XDG that were implemented to handle Amharic null subjects and objects, agreement of verbs with subjects and objects, and some of the special properties of relative clauses. Most of these extensions will also apply to other Semitic languages.

2. Extensible Dependency Grammar

As in other dependency grammar frameworks, XDG is lexical; the basic units are words and the directed, labeled dependency relations between them. In the simplest case, an analysis ("model" in XDG terms) of a sentence is a graph consisting of a set of dependency arcs connecting the nodes in the sentence such that each node other than the root node has a head and certain constraints on the dependencies are satisfied. As in some, but not all, other dependency frameworks, XDG permits analyses at multiple strata, known as **dimensions**, each corresponding to some level of grammatical abstraction. For example, one dimension could represent syntax, another semantics. Two dimensions may also be related by an explicit interface dimension which has no arcs itself but constrains how arcs in the related dimensions associate with one another. Debusmann includes a total of six simple dimensions and five interface dimensions in the English grammar discussed in his dissertation. In the general case, then, an analysis of a sentence is a multigraph consisting of a separate dependency graph for each dimension over a single sequence of word nodes. Figure 1 shows a possible analysis for the English sentence *John edited the paper* on two dimensions. The analysis follows the XDG convention of treating the end-of-sentence punctuation as the root of the sentence.

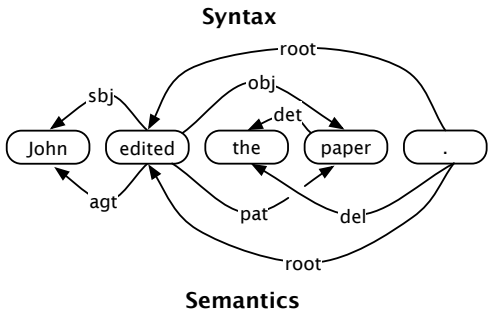

Figure 1: Two-dimensional XDG analysis of an English sentence. Arrows go from head to dependent. Words that do not participate in the semantic dimension are distinguished by delete arcs from the root node.

A grammatical analysis is one that conforms to a set of constraints, each generated by one or another **principle**. Each dimension has its own characteristic set of principles. Some examples:

- Principles concerned with the structure of the graph, for example, it may be constrained to be a tree or a directed acyclic graph.
- The Valency Principle, governing the labels on the arcs into and out of a given node.
- The Agreement Principle, constraining how certain features within some words must match features in other words.

- The Order Principle, concerned with the order of the words in the sentence.

As the framework is completely lexical, it is at the level of words or word classes that the principles apply. For example, the constraint that a finite present-tense verb in English must agree with its subject on the syntactic dimension could appear in the lexicon in this form:[1]

```
- gram: V_FIN_PRES
  syn:
    agree: [sbj]
```

The lexicon is organized in an inheritance hierarchy, with lexical entries inheriting attributes from their ancestor classes. For example, the verb *eats* would inherit the subject-verb agreement constraint from the V_FIN_PRES class.

Parsing and generation within the XDG framework take the form of constraint satisfaction. Given an input sentence to be parsed, lexicalization of the words invokes the principles that are referenced in the lexical entries for the words (or inherited from their ancestors in the lexical hierarchy). Each of these principle invocations results in the instantiation of one or more constraints, each applying to a set of variables. For example, a variable is associated with the label on the arc between two given nodes, and the domain for that variable is the set of possible arc labels that can appear on the arc. Among the constraints that apply to such a variable are those that are created by the Valency Principle. For example, for English transitive verbs, there is a valency constraint which requires that exactly one of the arcs leaving the verb must have an `obj` label. Constraint satisfaction returns all possible combinations of variable bindings, each corresponding to a single analysis of the input sentence.

The XDG framework has been applied to a number of languages, including a small fragment of Arabic (Odeh, 2004), but no one has yet addressed the complexities of morphosyntax that arise with Semitic languages. This paper represents a first effort.

3. Relevant Amharic Morphosyntax

3.1. Verb morphology

As in other Semitic languages, Amharic verbs are very complex (see Leslau (1995) for an overview), consisting of a stem and up to four prefixes and four suffixes. The stem in turn is composed of a root, representing the purely lexical component of the verb, and a template, consisting of slots for the root segments and for the vowels (and sometimes consonants) that are inserted around and between these segments. The template represents tense, aspect, mood, and one of a small set of derivational categories: passive-reflexive, transitive, causative, iterative, reciprocal, and causative reciprocal. For the purposes of this paper, we will consider the combination of root and derivational category to constitute the verb lexeme.

Each lexeme can appear in four different tense-aspect-mood (TAM) categories, conventionally referred to as perfect(ive), imperfect(ive), jussive/imperative, and gerund(ive). We represent verb lexemes in the lexicon in terms of the conventional citation form, the third person singular masculine perfective. For example, the verb *aywededm*[2] 'he is not liked' has the lemma *tewededе* 'he was liked', which is derived from the verb root *w.d.d*.

Every Amharic verb must agree with its subject. As in other Semitic languages, subject agreement is expressed by suffixes alone in some TAM categories (perfective and gerundive) and by a combination of prefixes and suffixes in other TAM categories (imperfective and jussive/imperative). Amharic is a null subject language; that is, a sentence does not require an explicit subject, and personal pronouns appear as subjects only when they are being emphasized for one reason or another.

An Amharic verb may also have a suffix representing the person, number, and gender of a direct object or an indirect object that is definite.[3] The corresponding suffixes in other Semitic languages are often considered to be clitics or even pronouns, but there are good reasons not to do so for Amharic. First, one or two other suffixes may follow the object suffix. Second, as with subjects, object personal pronouns may also appear but only when they are being emphasized. Thus we will consider Amharic to have optional object agreement as well as obligatory subject agreement and to be a null object as well as a null subject language.

3.2. Noun phrases

Amharic nouns without modifiers take suffixes indicating definiteness and accusative case for direct objects and prefixes representing prepositions:

hakim
doctor
'a doctor' (1)

hakimu
doctor-DEF
'the doctor' (2)

hakimun
doctor-DEF-ACC
'the doctor (as object of a verb)' (3)

lehakimu
to-doctor-DEF
'to the doctor' (4)

However, when a noun is modified by one or more adjectives or relative clauses, it is the first modifier that takes

[1] We use YAML syntax (`http://www.yaml.org/`) for lexical entries.

[2] Amharic is written using the Ge'ez script. While there is no single agreed-on standard for romanizing the language, the SERA transcription system, which represents Ge'ez graphemes using ASCII characters (Firdyiwek and Yaqob, 1997), is common in computational work on Amharic and is used in this paper. This transcription system represents the orthography directly, failing to indicate phonological features that the orthography does not encode, in particular, consonant gemination and the presence of the epenthetic vowel that breaks up consonant clusters.

[3] In the interest of simplification, indirect objects will be mostly ignored in this paper. Most of what will be said about direct objects also applies to indirect objects.

these affixes (Kramer, 2009). If a noun takes a determiner, the noun phrase needs no other indication of definiteness, but it is the determiner that takes the accusative suffix or prepositional prefix.

senefu *hakim*
lazy-DEF doctor
'the lazy doctor' (5)

lesenefu *hakim*
to-lazy-DEF doctor
'to the lazy doctor' (6)

yann *senef* *hakim*
that-ACC lazy doctor
'that lazy doctor (as object of a verb)' (7)

3.3. Relative clauses

Relative clauses in Amharic consist of a relative verb and zero or more arguments and modifiers of the verb, as in any clause. A relative verb is a verb in either the imperfective or perfective TAM with a prefix indicating relativization. As with a main clause verb, a relative verb must agree with its subject and may agree with its direct object if it has one. Both subjects and objects can be relativized.

yemiwedat *sEt*
REL-he-likes-her woman
'the woman that he likes' (8)

yemiwedat *wend*
REL-he-likes-her man
'the man who likes her' (9)

As noted above, when a noun is modified by a relative clause and has no preceding determiner, it is the relative clause that takes suffixes indicating definiteness or accusative case or prepositional prefixes.

yetemereqew *lj* *wendmE* *new*
REL-he-graduated-DEF boy my-brother is
'The boy who graduated is my brother.' (10)

yetemereqewn *lj* *alawqm*
REL-he-graduated-DEF-ACC boy I-don't-know
'I don't know the boy who graduated.' (11)

When a sequence of modifiers precedes a noun, it is the first one that takes the suffixes or prefixes.[4]

yetemereqew *gWebez* *lj*
REL-he-graduated-DEF clever boy
'the clever boy who graduated' (12)

Because the first modifier of a noun determines the syntactic role of the noun phrase in the clause as well as its definiteness, we will treat this modifier, rather than the noun, as the syntactic head of the noun phrase. There are at least two other reasons for doing this.

- The head noun of a noun phrase with an adjective or relative clause modifier is optional.

  tlqun *'merTalehu*
  big-DEF-ACC I-choose
  'I choose the big one.' (13)

  yemiwedat *alderesem*
  REL-he-likes-her he-didn't-arrive
  '(He) who likes her didn't arrive. (14)

  Headless relative clauses are found in many languages, for example, in the English translation of sentence (14). What makes Amharic somewhat unusual is that headless relative clauses and adjectives functioning as noun phrases can be formed by simply dropping the noun.

- Relative verbs agree with the main clause verbs that contain them. For example, in example (14) above, the third person singular masculine subject in the main clause verb agrees with the third person singular masculine subject of the relative clause verb.

Therefore we interpret relative clause modifiers as syntactic heads of Amharic nouns. Because XDG offers the possibility of one or more dimensions for semantics as well as syntax, it is straightforward to make the noun the semantic head, much as auxiliary verbs function as syntactic heads while the main verbs they accompany function as semantic heads in Debusmann's XDG grammar of English. This is discussed further below.

4. XDG for Amharic

In its current incomplete version, our Amharic grammar has a single layer for syntax and a single layer for semantics. The Syntax dimension handles word order, agreement, and syntactic valency.[5] The Semantics dimensions handles semantic valency.

Because the grammar still does not cover some relatively common structures such as cleft sentences and complement clauses, the parser has not yet been evaluated on corpus data.

4.1. Incorporating morphology

For a language like Amharic, it is impractical to list all wordforms in the lexicon; a verb lexeme can appear in more than 100,000 wordforms. Instead we treat the lexeme/lemma as the basic unit; for nouns this is their stem.[6]

[4] With two adjectives, both may optionally take the affixes (Kramer, 2009). We consider this to fall within the realm of coordination, which is not handled in the current version of the grammar described in this paper.

[5] Amharic word order is considerably simpler than that of a language such as English or German, and there are none of the problems of long-distance dependences in questions and relative clauses that we find in those languages. The only non-projective structures are those in cleft sentences and sentences with right dislocation, neither of which is handled in the current version of our grammar. In a later version, we will separate a projective linear precedence layer from a non-projective immediate dominance layer, as Debusmann does for English and German (2007).

[6] Unlike in most other Semitic languages, most Amharic nouns do not lend themselves to an analysis as template+root.

For verbs, as noted above, this is the root plus any derivational morphemes.

In parsing a sentence, we first run a morphological parser over each of the input words. We use the HornMorpho Amharic parser available at `http://www.cs.indiana.edu/~gasser/Research/software.html` and described in Gasser (2009). Given an Amharic word, this parser returns the root (for verbs only), the lemma, and a grammatical analysis in the form of a feature structure description (Carpenter, 1992; Copestake, 2002) for each possible analysis. For example, for the verb *ywedatal* 'he likes her', it returns the following (excluding features that are not relevant for this discussion):

```
'wededе', {'tam': 'impf',
           'rel': False,
           'sb': [-p1,-p2,-plr,-fem],
           'ob': [-p1,-p2,-plr,+fem]}
```

That is, it indicates that this a non-relative verb whose lemma is 'wededе' in imperfective TAM with a third person singular masculine subject and a third person singular feminine object.

It is this sequence of lemma-structure tuples rather than raw wordforms that is the input to the usual XDG lexicalization process that initiates parsing. We have not yet implemented generation, but the reverse process will occur there; that is, the output of constraint satisfaction will be a sequence of lemma-structure tuples which will then be passed to a morphological generator (also available in HornMorpho).

4.2. Null subjects and objects

XDG is grounded in the words occurring in a sentence, but it has to come to grips with the mismatch between nodes in different dimensions. For example, we probably do not want a strictly grammatical word such as *the* to correspond to anything at all on the semantic dimension. Debusmann handles the *deletion* of surface nodes using `del` arcs from the sentence root; this can be seen in the semantic dimension in Figure 1.

However, as far as we know, no one has addressed the reverse problem, that of nodes in some dimension which correspond to nothing on the surface. Null subjects and objects in a language such as Amharic present such a problem. They correspond to arguments that need to be explicit at the semantic level but are not present in the input to parsing. We are also working on a synchronous version of XDG with dimensions representing syntactic analyses in different languages. For a language pair such as Amharic-English, with Amharic as the input language, the nodes corresponding to English subject and object pronouns will have to come from somewhere.

We solve this problem by introducing "empty nodes" in the syntactic dimension. Each verb creates an empty node for its subject, and each transitive verb creates an additional one for its object. The nodes are used only when no explicit argument fills their role. We introduce a new XDG principle to handle these cases, the Empty Node Principle. When a word invoking this principle is found during lexicalization, a constraint is created which sanctions an arc from the verb with the relevant label (`sbj` or `obj`) to either an explicit word or the associated empty node, but not both. Figure 4.3. shows the analysis returned by our parser for the following sentence.[7]

yoHans *ywedatal*
Yohannis he-likes-her
'Yohannis likes her.' (15)

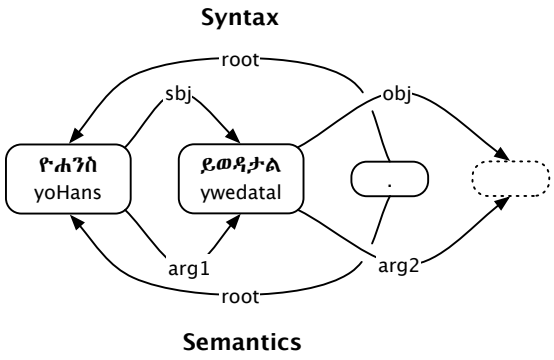

Figure 2: Empty nodes in Amharic. The transitive verb *ywedatal* 'he likes her' has no explicit object, so it is linked to an empty node by an `obj` arc in the Syntax dimension.

Note that our empty nodes are similar to the hidden nodes used in annotation for the Quranic Dependency Treebank project (Dukes et al., 2010).

4.3. Subject and object agreement

In the XDG grammars described by Debusmann and other researchers within the framework, agreement applies to two separate verb attributes. The `agrs` attribute is a list of possible features for the verb form, while the `agree` attribute is a list of arc labels for daughters which must agree with the verb. For example, the following could be part of the entry for the English very `eats`, representing the fact that this word has a single possibility for its agreement feature (third person singular) and the constraint that its subject must also be third person singular.

```
- word: eats
  syn:
    agrs: [3ps]
    agree: [sbj]
```

This limited approach to agreement fails to address the complexity of a language such as Amharic. First, the `agrs` attribute must distinguish subject, direct object, and indirect object features. Second, the `agree` attribute must specify which agreement feature of the mother verb agrees with the daughter on the specified arc. Third, the `agree` attribute must also allow for agreement with different features of the daughter when the daughter is verb itself, that is when it is the verb of a relative clause. Consider the entry for transitive verbs (actually a combination of several entries):

```
- gram: V_T
  syn:
    agree: {sbj: [sbj, [^,sbj,obj,iobj]],
            obj: [obj, [^,sbj,obj,iobj]]}
```

[7] In the Amharic dependency graphs in the figures we show the original Ge'ez forms that are the actual input to the parser as well as the transcribed forms.

This specifies that a transitive Amharic verb agrees with the words on both its outbound `sbj` and `obj` arcs, that the subject agrees with the `sbj` feature of the verb and the object agrees with the `obj` feature of the verb, and that the agreement feature of the daughter (subject or object) is either the whole word (denoted by `^`) or, in the case of a relative verb, its `sbj`, `obj` or `iobj` feature.
The following sentence is an example of a transitive verb whose subject and object features agree with nouns. The output of the parser on the Syntax dimension for this sentence is shown in Figure 3.

astEr *yoHansn* *twedewalec*
Aster Yohannis-ACC she-likes-him
'Aster likes Yohannis.' (16)

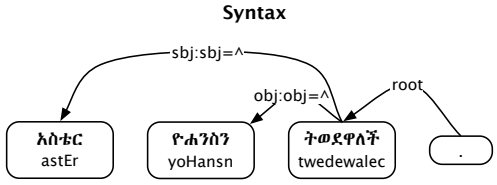

Figure 3: Simple subject-verb and object-verb agreement in Amharic. In addition to their arc labels, two arcs show mother and daughter features that agree. In these cases, the arc label precedes the colon, and the mother and daughter features are separated by "=".

Note that the verb agreement feature and the arc label need not be the same. For example, for an important subclass of Amharic verbs, the object suffix of the verb agrees with a syntactic argument that we will call the "topic", which does not take the accusative marker and is not the syntactic subject. In the following example, the verb's object suffix is third person singular feminine, agreeing with the nominative topic *astEr*.

astEr *dekmWatal*
Aster it-has-tired-her
'Aster is tired.' (17)

The verb in this sentence, *dekeme* 'tire', has the following in its entry:

```
- lexeme: dekeme
  syn:
    agree: {obj: [top, [^,sbj,obj,iobj]]}
```

Figure 4 shows the parser's analysis of sentence (17).

4.4. Relative clauses

As argued above, relative verbs are best treated as the heads of their noun phrases. When a relative verb has a head noun, the verb's subject, object, or indirect object feature must agree with that noun, depending on the role it plays in the verb's argument structure. In our grammar, we join the relative verb to its head noun in the Syntax dimension by an arc with a label specifying this role, that is, `sbj`, `obj`, or `iobj`. Since verbs are already constrained to agree with their arguments, the agreement between the relative verb

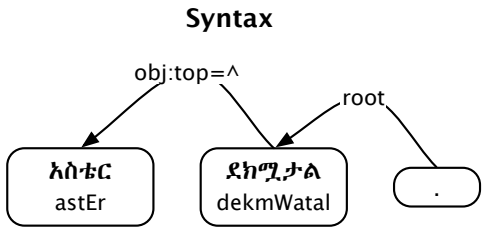

Figure 4: Agreement of a topic with a verb's object suffix.

and the noun it modifies does not need to be stated separately in the grammar. For illustration, however, we show what this constraint would look like in the entry for object relative verbs.

```
- gram: V_REL_OBJ
  syn:
    agree: {obj: [obj, ^]}
```

Sentence (18) is an example of a sentence with an object relative clause. The analysis of the sentence by our system on the Syntax dimension is shown in Figure 5. The object feature of the relative verb *yemtTelaw* 'that she hates him' agrees with the modified noun *wendlj* 'boy'; both are third person singular masculine. Two other agreement constraints are also satisfied in this sentence. The subject feature of the main verb *tameme* 'he-got-sick' agrees with the object feature of the relative verb; both are third person singular masculine. The subject feature of the relative verb agrees with its subject *astEr*; both are third person singular feminine.

astEr *yemtTelaw* *wendlj* *tameme*
Aster REL-she-hates-him boy he-got-sick
'The boy that Aster hates got sick.' (18)

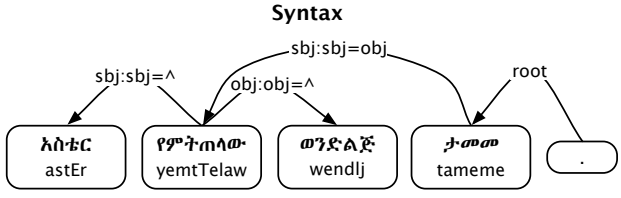

Figure 5: Syntactic analysis of a sentence with a relative clause.

We model the semantics of a sentence with a relative clause as a directed acyclic graph in which the shared noun has multiple verb heads. The relative clause predicate is distinguished from the main clause predicate by a `rel` rather than a `root` arc into it from the sentence root. Figure 6 shows the analysis of sentence (18) on the Semantics dimension.

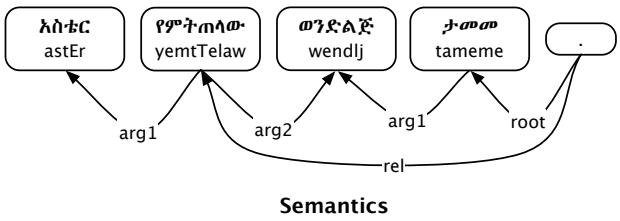

Figure 6: Semantic analysis of a sentence with a relative clause.

Relative clauses without nouns have no overt form corresponding to the shared semantic argument, so we introduce this argument as an empty node. Sentence (19) is sentence (18) with the noun *wendlj* 'boy' dropped. The analyis of this sentence is shown in Figure 7.

astEr	*yemtTelaw*	*tameme*
Aster	REL-she-hates-him	he-got-sick

'The one that Aster hates got sick.' (19)

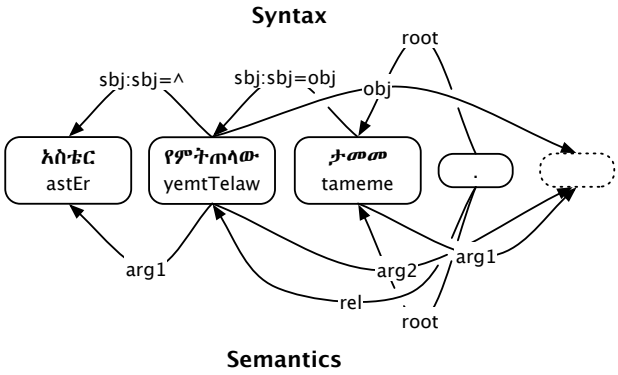

Figure 7: Analysis of a relative clause with no modified noun.

Without further constraints, however, the grammar assigns multiple analyses to some sentences and parses some ungrammatical sentences with relative clauses. Consider the following ungrammatical sentence.

**astEr*	*yemtTelaw*	*wendlj*	*tamemec*
Aster	REL-she-hates-him	boy	she-got-sick

'The boy that Aster hates (she) got sick.' (20)

This satisfies the constraint that subject of the main verb *tamemec* agree with some feature of the relative verb (its subject) and the constraint that the some feature of the relative verb (its object) agree with the modified noun *wendlj*. To exclude sentences like this, we need a further XDG principle, which we call the Cross-Agreement Principle. This specifies a fundamental fact about relative clauses in all languages, that the same noun functions as an argument of two different verbs, the main clause verb and the relative verb. The Cross-Agreement Principle forces the same feature of the relative verb to agree with the main clause verb and the modified noun. By this principle our parser finds no analysis for sentence (20) because the feature of the relative verb *yemtTelaw* that agrees with the modified noun (its object) differs from the feature that agrees with the main verb (its subject). This is illustrated in Figure 8. The grammar fails to parse this sentence between the features marked with red boxes do not agree.

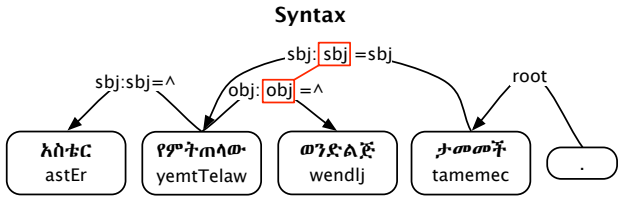

Figure 8: Violation of the Cross-Agreement Principle. The features in red boxes should match.

5. Conclusions

This paper has described an implementation of Extensible Dependency Grammar for the Semitic language Amharic. Amharic is interesting because it suffers from a serious lack of computational resources and because its extreme morphological complexity and elaborate interactions of morphology with syntax present challenges for computational grammatical theories. Besides the strongly lexical character that it shares with other dependency grammar frameworks, XDG is attractive because of the modularity offered by separate dimensions. We have seen how this modularity permits us to handle the agreement constraints on a relative verb by treating such verbs as the heads of noun phrases on the Syntax, but not the Semantics dimension. We have also seen that XDG requires some augmentation to deal with null subjects and objects and the intricacies of verb agreement. These complexities of Amharic are not unique. Much of what has been said in this paper also applies to other Ethio-Semitic languages such as Tigrinya. In addition to expanding the coverage of Amharic, further work on this project will be directed at developing synchronous XDG grammars to support translation between the different Semitic languages spoken in Ethiopia and Eritrea.

6. References

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA, USA.

Ralph Debusmann, Denys Duchier, and Geert-Jan M. Kruijff. 2004. Extensible dependency grammar: A new methodology. In *Proceedings of the COLING 2004 Workshop on Recent Advances in Dependency Grammar*, Geneva/SUI.

Ralph Debusmann. 2007. *Extensible Dependency Grammar: A Modular Grammar Formalism Based On Multigraph Description*. Ph.D. thesis, Universität des Saarlandes.

Kais Dukes, Eric Atwell, and Abdul-Baquee M. Sharaf. 2010. Syntactic annotation guidelines for the Quranic Arabic treebank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta.

Yitna Firdyiwek and Daniel Yaqob. 1997. The system for Ethiopic representation in ASCII. URL: citeseer.ist.psu.edu/56365.html.

Michael Gasser. 2009. Semitic morphological analysis and generation using finite state transducers with feature structures. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 309–317, Athens, Greece.

Ruth Kramer. 2009. *Definite Markers, Phi Features, and Agreement: a Morphosyntactic Investigation of the Amharic DP*. Ph.D. thesis, University of California, Santa Cruz.

Wolf Leslau. 1995. *Reference Grammar of Amharic*. Harrassowitz, Wiesbaden, Germany.

Marwan Odeh. 2004. Topologische dependenzgrammatik fürs arabische. Technical report, Saarland University. Forschungspraktikum.

A Syllable-based approach to verbal morphology in Arabic

Lynne Cahill
University of Brighton
NLTG, Watts Building, Lewes Rd, Brighton BN2 4GJ, UK
E-mail: L.Cahill@brighton.ac.uk

Abstract

The syllable-based approach to morphological representation (Cahill, 2007) involves defining fully inflected morphological forms according to their syllabic structure. This permits the definition, for example, of distinct vowel constituents for inflected forms where an ablaut process operates. Cahill (2007) demonstrated that this framework was capable of defining standard Arabic templatic morphology, without the need for different techniques. In this paper we describe a further development of this lexicon which includes a larger number of verbs, a complete account of the agreement inflections and accounts for one of the oft-cited problems for Arabic morphology, the weak forms. Further, we explain how the use of this particular lexical framework permits the development of lexicons for the Semitic languages that are easily maintainable, extendable and can represent dialectal variation.

1. Introduction

The Semitic languages are linguistically interesting for a number of reasons. One of the most widely discussed aspects of these languages is the so-called templatic morphology with the typical triliteral verbal (and nominal) roots and their vocalic inflections. In the 1980s a rash of studies emerged discussing ways of describing this morphology and associated problems such as spreading (where only two consonants are specified in the root) and the weak verbs, where one of the consonants in the root is one of the "weak" consonants or glides, *waw* (/w/) or *yaa* (/j/).

Cahill (2007) presented an alternative to these approaches which made use of a framework developed to describe European languages which is based on defining the syllabic structure for each word form. The lexicon is defined as a complex inheritance hierarchy. The fundamental assumption behind this work is that the vocalic inflections can be defined in exactly the same way as an ablaut process commonly seen in European languages. Even the less obviously similar derivations which involve "moving around" of the root consonants (for the different binyan[1] derivations) can be dealt with using the same apparatus as required for consonant adaptations in European languages.

The account in Cahill (2007) describes the basic lexical hierarchy for triliteral verbal roots in MSA with a single verb root being used to demonstrate the ability to generate the full (potential) range of forms with the framework. The account does not cover the agreement inflections (the prefixes and suffixes), nor does it cover anything other than verbs with triliteral strong roots. In this paper we present the latest extensions to this work, which aims ultimately to provide a complete account of the verbal and nominal morphology of Modern Standard Arabic (MSA).

The key developments we report here are:

1. the addition of the agreement inflections;
2. the addition of the apparatus required for handling non-standard roots.

The first of these does not amount to anything very different from a large number of accounts of affixal morphology within an inheritance framework. The second is more interesting, but turns out to be no more challenging for the framework than various types of phonological conditioning in the morphological systems of many European languages. We illustrate our approach to the weak roots with an analysis of one particular weak root, the defective root *r-m-j*, "throw", which has a weak final consonant.

Finally, we discuss the ways in which the framework presented allows for easier extension of the lexicons to enable the development of large-scale lexical resources for the Arabic languages, and how the lexicon structure will permit the definition of dialects in addition to the current account of MSA.

2. MSA verbal morphology

The verbal morphology of the semitic languages has attracted plenty of attention in both the theoretical and computational linguistics communities. What makes it interesting, particularly from the perspective of those exposed only to European languages, is the structure of the stems, involving consonantal roots, vocalic inflections and templates or patterns defining how the consonants and vowels are ordered. Several approaches to the task have been implemented, most based to some degree on the two-level morphology of Koskenniemi (1983), although once adapted to allow for the formation of semitic roots, it ended up being four-level morphology

[1] We use the Hebrew term "binyan" to refer to the different derived forms, also known as "measures" or "forms".

(see e.g. Kiraz (2000)).

The stem formation has already been shown (Cahill, 2007) to be elegantly definable using an approach which was developed mainly for defining European languages such as English and German. We will describe this technique in the next section. However, semitic morphology, and specifically the morphology of MSA, involves other word formation and inflection processes. One of the areas that has attracted a good deal of attention is the issue of what happens when the verb root, traditionally assumed to consist of three consonants, does not fit this pattern. The three principal situations where this happens are in the case of biliteral or quadriliteral roots, where there are either two or four consonants instead of the expected three, and the weak roots, where one of the consonants is a "weak" glide, i.e. either /w/ or /j/.

Where a root has only two consonants, one or other of those consonants is used as the third (middle) consonant, which one depending on the stem shape. Where a root has four consonants, the possible forms are restricted to forms where there are at least four consonant "slots". Early accounts of these types of root include a range of means of "spreading" where post lexical processes have to be invoked to copy one or other of the consonants (see, e.g. Yip (1988)).

The issue of bi- and quadri-literal roots is relatively simply handled within the syllable-based framework, as described in section 4 below. The weak roots are slightly more complex, but nevertheless amenable to definition in a similar way to the kind of phonological conditioning seen, for example, in German final consonant devoicing, where the realisation of the final consonant of a stem depends on whether it is followed by a suffix beginning with a vowel or not. The Syllable-based Morphology framework has been developed to allow for the realisation of fully inflected forms to be determined in part by phonological characteristics of the root or stem in question. This means that, while Arabic weak roots are often cited as behaving differently **morphologically**, we argue that they behave entirely regularly morphologically, but their behaviour is determined by their phonology.

3. Syllable-based morphology

The theory of syllable-based morphology (SBM) can trace its roots back to the early work of Cahill (1990). The initial aim was to develop an approach to describing morphological alternation that could be used for all languages and all types of morphology. Cahill's doctoral work included a very small indicative example of how the proposed approach could describe Arabic verbal stem formation. The basic idea behind syllable-based morphology is simply that one can use syllable structure to define all types of stem alternation, including simple vowel alternations such as ablauts. All stems are defined by default as consisting of a string of tree-structured syllables. Each syllable consists of an onset and a rhyme and each rhyme of a peak and a coda[2]. The simplest situation is where all wordform stems of a particular lexeme are the same. In this case, we can simply specify the onsets, peaks and codas for all of the syllables. For example, the English word "pit" has the root /pIt/ and this is also its stem for all forms (singular, plural and possessive). The phonological structure of this word in an SBM lexicon would therefore by defined as follows[3]:

```
<phn syl1 onset> == p
<phn syl1 peak> == I
<phn syl1 coda> == t
```

This example is monosyllabic, but polysyllabic roots involve identifying individual syllables by counting from either the left or right of a root. For suffixing languages, the root's syllables are counted from the right, while for prefixing languages, they are counted from the left. For Arabic, although both pre- and suffixing processes occur, the decision has been made to count from the right, as there is more suffixation. However, as the roots in Arabic, to all intents and purposes, always have the same number of syllables, it is not important whether we choose to call the initial syllable syl1 or syl2.

In the case of simple stem alternations such as ablaut, the peak of a specified syllable is defined as distinct for the different wordforms. That is, the realisation of the peak is determined by the morphosyntactic features of the form. To use a simple example, for an English word *man*, which has the plural *men*, we can specify in its lexical entry:

```
<phn syl1 peak sing> == a
<phn syl1 peak plur> == E.
```

As the individual consonants and vowels are defined separately for any stem, the situation for Arabic is actually quite straightforward. For each verb form, inflected or derived, the consonants and vowels are defined, not in terms of their position in a string or template, but in terms of their position in the syllable trees. Thus, Cahill (2007) describes how the three consonants can be positioned as the onset or coda of different syllables. The vowels are defined in terms of tense/aspect.

Figure 1 shows how the (underspecified) root structure for the root *katab* looks. This is defined in DATR as follows[4]:

```
<phn syl2 onset> == Qpath:<c1>
```

[2] The term "peak" is used to refer to the vowel portion of the syllable, rather than the sometimes used "nucleus". The syllable structure is relatively uncontroversial, having been first proposed by Pike and Pike (1947).

[3] We use the lexical representation language DATR (Evans and Gazdar, 1996) to represent the inheritance network and use SAMPA (Wells, 1989) to represent phonemic representations.

[4] This is specified at the node for verbs, which defines all of the information that is shared, by default, by all verbs in Arabic.

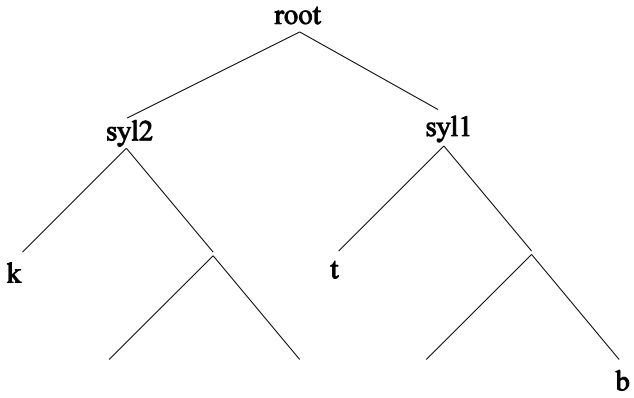

Figure 1: the structure of /katab/

```
<phn syl1 onset> == Qpath:<c2>
<phn syl1 coda>  == Qpath:<c3>
```

These equations simply say that (by default) the onset of the initial syllable is filled by the first consonant (*c1*), the onset of the second syllable is filled by the second consonant (*c2*) and the coda of the second syllable is filled by the third consonant (*c3*). The precise position of the consonants depends not only on the binyan, but also on tense. By default, the past tense has the structure in figure 1, but the present tense has that in figure 2.

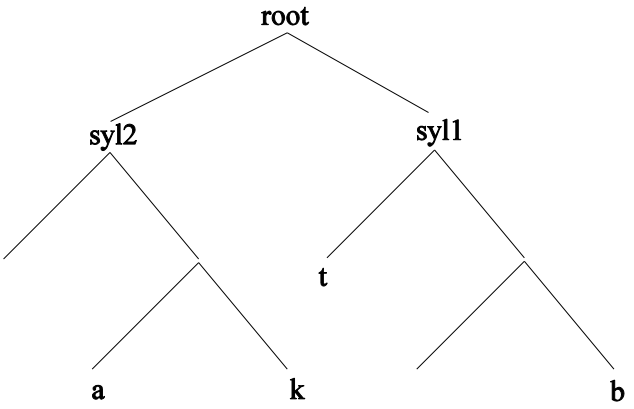

Figure 2: the structure of /aktub/

Affixation is handled as simple concatenation, such that (syllable-structured) affixes concatenate with (syllable-structured) stems to make longer strings of structured syllables. For a simple case such as English noun plural suffixation, for example, we need to specify that a noun consists of a stem and a suffix. We then need to state that, by default, the suffix is null, and that in the case of the plural form, a suffix is added.

```
<mor word form> ==
       "<phn root>" "<mor suffix>"
<mor suffix> == Null
<mor suffix plur> ==
       Suffix_S:<phn form>
```

As we are dealing with phonological forms, we also need to specify how the suffix is realised, which is defined at the separate "Suffix_S" node[5].

One of the key aspects of SBM is that all forms are defined in terms of their syllable structure. This does lead to a slight complication with affixes which consist of a single consonant, for example. The SBM approach to this is to say that there is a necessary post-lexical resyllabification process which takes place after all affixes have been added and so it is not a problem to define affixes as (at least) single syllables, even if they are syllables with no peaks. Although this may seem a little counter-intuitive, the issue of resyllabification is clearly one which must be addressed. If we affix *–ed* (/Id/) to an English verb stem which ends in a consonant, it is almost always the case that that consonant becomes the onset of the suffix syllable, while it is the coda of the final syllable of the stem if no affix is added. Indeed, in most languages it is even the case that resyllabification takes place across word boundaries in normal speech.

4. Extensions to the framework

Cahill's (2007) account of Arabic morphology only covered the stem formation, and did not attempt to cover anything other than straightforward triliteral strong verb roots. In fact, the fragment published in the appendix of that paper includes a single example verb entry, an example of a standard strong triliteral verb. In this section we discuss the three ways in which we have, to date, extended the lexicon.

4.1 Adding more lexemes

We have extended the lexicon initially to include a larger number of strong, triliteral verbs. This is an extremely simple process in the lexicon structure provided as all that needs to be specified are the three consonants in the root. This does result in overgeneration, as all possible stems, for all binyanim, are generated. However, it is a simple process to block possible forms, and there is a genuine linguistic validity to the forms, such that, if a particular verb has a Binyan 9 form, then we know what form it will take.

The issue of how many binyanim to define is an interesting one, and one we will come back to in the discussion of extending coverage to dialects of Arabic. Classical Arabic has a total of fifteen possible binyanim, while MSA makes use of ten of these standardly and two more in a handful of cases.

4.2 Agreement inflections

The next extension to the existing lexicon was to add the agreement inflections. These include prefixes and suffixes and mark the person, number and gender of the form. As noted above, the affixal inflections do not pose any particular difficulties for the syllable-based framework.

[5] For more detail of this type of SBM definition for German, English and Dutch, see Cahill and Gazdar (1999a, 1999b).

The "slots" for the affixes were already defined in the original account, so it was simply a case of specifying the realisations. The exact equations required for this will not be covered in detail here, but we note that the affixes display typical levels of syncretism and default behaviour so that, for example, we can specify that the default present tense prefix is *t-* as this is the most frequent realisation, but the third person present tense prefix is *y-* while the third person feminine prefix is *t-*. This kind of situation occurs often in defining default inheritance accounts of inflection and is handled by means of the following three equations[6]:

```
<agr prefix pres> == t
<agr prefix pres third> == y
<agr prefix pres third femn> == t
```

4.3 Non-standard verb roots

The final extension which we report in this paper is the adaptation of the framework for stem structure to take account of the different types of verb root, as discussed in section 2.

Dealing with biliteral roots involves specifying for each consonant (i.e. onset or coda) defined in the stem structure whether it should take the first or third consonant value, **if the second consonant is unspecified**. Thus, biliteral roots have their second consonants defined thus:

```
<c2> == Undef
```

Then, an example of defining the correct consonant involves a simple conditional statement:

```
<phn syl2 onset> ==
    IF:<EQUAL:<"<c2>" Undef>
    THEN "<c1>"
    ELSE "<c2>"
```

This simply states that, if the second consonant is unspecified, then the first consonant takes this particular position, but if not, then the second consonant will take its normal place. In positions where the absent second consonant is represented by the third consonant simply require the third line above to give *c3* rather than *c1* as the value.

In order to handle quadrilateral roots, we need a separate node for these verbs which defines which of the consonants occupies each consonant slot in the syllable trees. In many cases these are inherited from the Verb node, for example, the first consonant behaves the same in these roots. Typically, where a triliteral root uses *c1*, a quadriliteral root will use *c1*; where a triliteral root uses *c3*, the quadriliteral root will use *c4* in most cases, but *c3* in others; where a triliteral root uses *c2*, the quadriliteral root will either use *c2* or *c3*, so these equations have to be specified.

The weak roots have a glide in one of the consonant positions. This leads to phonologically conditioned variation from the standard stem forms. For example, the hollow verb *zawar* ("visit") has a glide in second consonant position. This leads to stem forms with no middle consonant, and a *u* in place of the two *a*s. In order to allow for this variation, we need to check whether the second consonant is a glide and this will determine the realisations. This check must be done for each onset, peak and coda that is defined as having possible variation, and involves a simple check whether the second consonant is a /w/ or a /j/. In each case the behaviour is the same for the consonant itself, i.e. it is omitted, but different for the vowel. With /w/, the vowel is /u/ but with /y/ it is /i/, in the second vowel position.

There are two possible approaches we could take to defining the different behaviour of weak verbs. The first is to specify a finite state transducer to run over the default forms. For example, we could state that if a verb root has the sequence /awa then this becomes /u:/. The second approach is to define the elements of the syllable structure according to the phonological context in which they occur. We opt for the second of these approaches for a number of reasons. The first is that we wish to minimise the different technologies used in our lexicon. Although FSTs are very simple to implement, we want to resist using them if possible, in order to make use only of the default inheritance mechanisms available to us. The second is that we are not yet at a stage in the project where we have enough varied data for all of the different verb and noun forms to be certain that any transducer we devise will not over apply, whereas we can be more confident of the specific generation behaviour of the inheritance mechanisms we are employing in the lexicon structure as a whole.

One disadvantage of the approach we have chosen to take is that is does result in somewhat more complex definitions in our lexical hierarchy. For example, if we only define strong triliteral verb roots, then our lexical hierarchy can include statements like:

```
<phn syl1 onset> == Qpath:<c1>
```

which are very simple. If we include all of the variation in this hierarchy then we need more statements (to distinguish between, for example, past and present tense behaviour) and those statements are more complex. This is because, even for the standard strong triliteral roots, we need to check for each consonant whether or not it is weak and for each vowel, we need to check whether it is adjacent to a weak consonant . For this reason, we do not include the DATR code which defines the weak verb forms, but rather describe the checks needed.

The approach we take involves two levels of specification. At the first level, each equation defining a consonant or a vowel calls on a simple checking function to determine

[6] We have specified the present tense prefixes without the /a/, as this is present in all forms. We therefore consider that this segment is part of the present tense stem.

whether the realisation is the default one or something different. These calls to checking functions may take different arguments. Thus, the simplest type just needs to be passed the root consonant in question and will determine whether it is realised (if it is strong) or not (if it is weak). In more complex situations, e.g. where a weak root has /u:/ where it would by default have /awa/, we need to pass both the consonant and at least one of the vowels.

The checking nodes are each very simple. The simplest just state that the consonant is realised if it is strong but not if it is weak:

```
Check_ajwaf_cons:
    <$weak> ==
    <$strong> == $strong.
```

We add similar checks to the equations for vowels so that, instead of the default stem form of /zawar/ we get the correct (first and second person[7]) stem of /zur/. The other weak forms involve similar checks for the other consonants.

These checks are very similar to the checks we can see in the syllable-based accounts of, for example, German (Cahill and Gazdar, 1999a). The realisation of the final consonant in any stem in German is dependent on whether or not there is a suffix which begins with a vowel. Therefore, the equation specifying that consonant checks for the beginning of the suffix (if there is one) and for underlying voiced consonants returns the voiced variant only if there is a vowel following, and returns the voiceless variant otherwise.

To clarify the entire process involved in generating a verb form from our lexicon, we shall now describe the derivation of the present tense active third person plural masculine form of the verb "throw" (they(m) throw). This is a weak (defective) verb, with a root of *r-m-j*. The first thing we do is look for the agreement prefix. Our Verb node tells us that this is /j/. Next we need to determine the stem for this form. The stem is defined as having /a/ as the peak of the first syllable (the default value for all present tense forms) and the first consonant of the root, i.e. /r/ as the coda of the first syllable. We determine this by checking whether it is weak or not. Once we have determined that it is a strong consonant, it takes its place in the syllable structure. The onset of the second syllable is the second consonant, in this case /m/, just as it is in most stems. Once again, we check that this is not a weak consonant before placing it in its position. At this point we start to find different behaviour. If the final consonant was strong then we would get a /u/ as the peak of the second syllable. However, as the final consonant, /j/ is weak, the peak is null. Similarly, the final consonant is not realised, because it is weak. So, our stem is fully realised as /arm/. Finally, we look for the agreement suffix, which is defined as /u:na/. So, our fully inflected form is /jarmu:na/. The syllable structure of the stem is shown in figure 3.

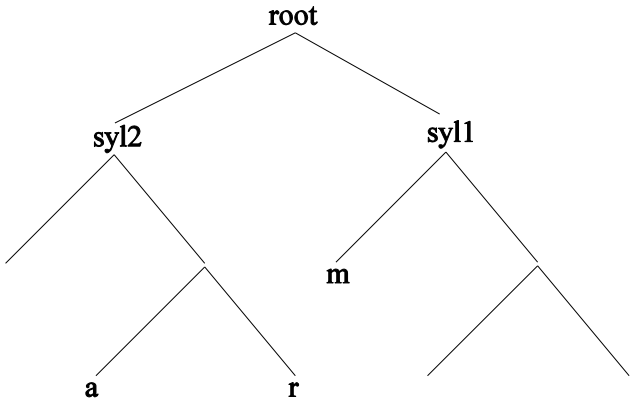

Figure 3: the structure of /arm/

5. Future directions

The extensions we report on here are only the start of a program of research which will add nouns and other non-regular morphological forms (e.g. the broken plural). The project is also going to add orthographic forms, derived from the phonological and morphological information, and supplement these with information about the relative frequency of the ambiguous forms in unvocalised script.

5.1 Extension of the lexicon

The DATR implementation of the lexicon is based on the lexicon structure of PolyLex (Cahill and Gazdar, 1999b). This gives the lexicon two big advantages over other lexicon structures. The first is the default inheritance machinery, which allows very easy extension. It is extremely easy to add large numbers of new lexemes automatically, as long as the hierarchy defines all of the variation. The task is simply to add basic root information (the consonants and the meaning and any irregularities peculiar to that lexeme – although there should not be many irregularities in new additions, as the most frequent words will have been added, and it is usually the more frequent words which are irregular) and choose the node in the hierarchy from which it should inherit. The PolyLex project developed tools to allow the generation of large numbers of additional lexical entries from a simple database format which includes the important information.

Crucially, the use of default inheritance means that, even if we do not have all of the information available to determine the exact morphological behaviour of a particular lexeme, we can assign sensible default values. For example, if we wanted to add a new English noun to our lexicon, and we have not seen an example of that noun

[7] The discussion here has been simplified for the sake of brevity. The first and second person stem forms are the same, and are defined here, but the third person stems are different. This is not a problem for our account, as the framework is specifically designed to allow both morphosyntactic and phonological information to be used in determining the correct form.

in its plural form, we can add it as a regular noun, and generate a plural form which adds the *–s* suffix. This may not be correct, but it is a reasonable guess, and the kind of behaviour we would expect from a human learning a language. This is useful if the data we use to extend our lexicons comes, for example, from corpora – often a necessity for languages which do not have large established resources.

In terms of the Arabic lexicon we describe here, the forms of verbs, even those with weak roots, do not need any further specification, as the lexical hierarchy defines the alternation in terms of the phonological structure of the root. Therefore, if a newly added root has a weak consonant, the correct behaviour will automatically be instigated by the recognition of that weak consonant.

This process has already been tested with a random selection of 50 additional strong verbs, two weak verbs for each of the consonant positions (i.e. two with weak initial consonants, two with weak medial consonants and two with weak final consonants) and one with two weak consonants. The resulting forms for some of these verbs are included in Appendix 2.

5.2 Adding more dialects

Another issue which causes much concern in the representation and processing of Arabic is the question of the different varieties or dialects. Buckwalter (2007) says "... the issue of how to integrate morphological analysis of the dialects into the existing morphological analysis of Modern Standard Arabic is identified as the primary challenge of the next decade." (p. 23). Until relatively recently, the issue of dialects in Arabic was only relevant for phonological processing, as dialects did not tend to be written. However, the rapid expansion of the Internet, amongst other developments, means that written versions of the various dialects are increasingly used, and processing of these is becoming more important.

The PolyLex architecture was developed as a multilingual representation framework, particularly aimed at representing closely related languages (the PolyLex lexicons themselves include English, German and Dutch). The framework involves making use of extended default inheritance to specify information which is shared, by default, by more than one language, with overrides being used to specify differences between languages as well as variant behaviour within a single language (such as irregular or sub-regular inflectional forms). In the case of English, German and Dutch, for example, it is possible to state that, by default, nouns form their plural by adding an *–s* suffix. This is true of all regular nouns in English and of one class of nouns in both Dutch and German. Importantly, those classes in Dutch and German are the classes that new nouns tend to belong to, so assuming that class to be the default works well.

One of the great advantages of such a framework is that, being designed to work for closely related languages, it is also appropriate for dialects of a single language. We can map the situation for MSA[8] and the dialects onto this directly, with MSA taking the place of the multilingual hierarchy and the dialects taking the place of the separate languages here. The assumption is that, by default, the dialects inherit information (about morphology, phonology, orthography, syntax and semantics) from the MSA hierarchy, but any part of that information can be overridden lower down for individual dialects. There is nothing to prevent a more complex inheritance system, for example, to allow two dialects to share information below the level of the MSA hierarchy, but to also specify some distinct bits of information.

6. Conclusions

The approach to Arabic morphology presented here is still in the early stages of development. It does, nevertheless, demonstrate a number of crucial points. First, it backs Cahill (2007) in showing that the SBM approach appears to be adequate to define those aspects of Arabic morphology that have frequently been cited as problematic. It is important to establish proof of concept in employing a new approach to specifying the morphology of any language, and the (admittedly small) lexicon does demonstrate the possibility of handling bi- and quadriliteral roots as well as weak verb roots within the SBM framework. Although not all of the details for all of the verbal morphology have yet been implemented, nothing has been shown to cause any significant difficulties that cannot be overcome in the framework.

Secondly, having established that the approach appears to be feasible for the complexities of Arabic morphology, it follows that the implementation of the morphology in the form of a PolyLex-style lexicon will permit the definition of dialectal variation, thus allowing the development of a full lexicon structure defining MSA, Classical Arabic as well as regional variants in an efficient and practically maintainable way. Although the details remain to be worked out, the assumed structure would involve a core lexicon which defines, for example, all fifteen of the Classical Arabic binyanim, with each of the lexicons for a "real" language specifying which of those are employed within that language or dialect.

The PolyLex lexicon structure allows the definition of defaults, which can be overridden at any of a number of levels. It is possible to override some pieces of information for an entire language or dialect, for a word-class such as nouns, for a sub-class of nouns or verbs or for an individual lexeme. This makes it very efficient at representing lexical information which tends

[8] It may prove more accurate and useful to have Classical Arabic in the multilingual position, as this probably includes more of the range of forms that the different dialects would need to inherit.

to be very largely regular. It also makes it very easy to add new lexemes, even if it has not been wholly established what all of the correct forms of that lexeme are. To use an analogy from child language acquisition, a child hearing an English noun, will assume that its plural is *–s* unless and until they hear an irregular plural form for it. Similarly, a child learning Arabic will assume that a new verb it hears follows the default, regular patterns unless and until they hear non-regular forms. That is the kind of behaviour that our default inheritance lexicon models when adding new lexemes.

7. Acknowledgements

The work reported here was undertaken as part of the ESRC (UK) funded project *Orthography, phonology and morphology in the Arabic lexicon*, grant number RES-000-22-3868. Their support is gratefully acknowledged. I am also grateful to the anonymous reviewers for their constructive comments.

8. References

Buckwalter, Tim. (2007) Issues in Arabic Morphological Analysis. In Abdelhadi Soudi, Antal van der Bosch and Günther Neumann (eds.) *Arabic Computational Morphology* Dordrecht : Springer. pp. 23-41.

Cahill, Lynne. (2007) A Syllable-based Account of Arabic Morphology. In Abdelhadi Soudi, Antal van der Bosch and G ü nther Neumann (eds.) *Arabic Computational Morphology* Dordrecht : Springer. pp. 45-66.

Cahill, Lynne. (1990) Syllable-based Morphology. *COLING-90*, Vol 3, pp. 48-53, Helsinki.

Cahill, Lynne and Gazdar, Gerald. (1999a) German noun inflection. *Journal of Linguistics*, 35 :1, pp. 211-245.

Cahill, Lynne and Gazdar, Gerald. (1999b) The PolyLex architecture : multilingual lexicons for related languages. *Traitement Automatique des Langues*, 40 :2, pp. 5-23.

Evans, Roger and Gazdar, Gerald. (1996) DATR : a language for lexical knowledge representation. *Computational Linguistics*, 22 :2, pp. 167-216.

Kiraz, George. (2000) A Multi-tiered Non-linear Morphology using Multi-tape Finite State Automata : A Case Study on Syriac and Arabic. *Computational Linguistics*, 26 :1, pp. 77-105.

Koskenniemi, Kimmo. (1983) *Two-level Morphology : A General Computational Model for Word-form Recognition and Production.* PhD Dissertation University of Helsinki.

Pike, Kenneth L. and Pike, Eunice V. (1947) Immediate constituents of Mazateco syllables. *International Journal of American Linguistics*, 13, pp. 78-91.

Wells, John. (1989) Computer-coded phonemic notation of individual languages of the European Community. *Journal of the International Phonetic Association*, 19 :1, pp. 31-54.

Yip, Moira. (1988) Template Morphology and the Direction of Association. *Natural Language and Linguistic Theory*, 6.4. pp. 551-577.

Appendix: Sample output

The DATR-implemented lexicon can be compiled and queried. In this appendix, we include the full lexical dumps for three lexemes: the fully regular strong triliteral, *k-t-b*, "write"; the weak (defective) verb *r-m-y*, "throw"; and the "doubly" weak verb *T-w-y*, "fold". The dumps give the present and past active forms for the first binyan.

```
Write:<bin1 mor word past act first sing>
        = k a t a b t u.
Write:<bin1 mor word past act first plur>
        = k a t a b n a:.
Write:<bin1 mor word past act secnd sing
        masc> = k a t a b t a.
Write:<bin1 mor word past act secnd sing
        femn> = k a t a b t i.
Write:<bin1 mor word past act secnd plur
        masc> = k a t a b t u m.
Write:<bin1 mor word past act secnd plur
        femn> = k a t a b t u n n a.
Write:<bin1 mor word past act third sing
        masc> = k a t a b a.
Write:<bin1 mor word past act third sing
        femn> = k a t a b a t.
Write:<bin1 mor word past act third plur
        masc> = k a t a b u:.
Write:<bin1 mor word past act third plur
        femn> = k a t a b n a.
Write:<bin1 mor word pres act first sing>
        = a k t u b u.
Write:<bin1 mor word pres act first plur>
        = n a k t u b u.
Write:<bin1 mor word pres act secnd sing
        masc> = t a k t u b u.
Write:<bin1 mor word pres act secnd sing
        femn> = t a k t u b i: n a.
Write:<bin1 mor word pres act secnd plur
        masc> = t a k t u b u: n a.
Write:<bin1 mor word pres act secnd plur
        femn> = t a k t u b n a.
Write:<bin1 mor word pres act third sing
        masc> = j a k t u b u.
Write:<bin1 mor word pres act third sing
        femn> = t a k t u b u.
Write:<bin1 mor word pres act third plur
        masc> = j a k t u b u: n a.
Write:<bin1 mor word pres act third plur
        femn> = j a k t u b n a.

Throw:<bin1 mor word past act first sing>
        = r a m a j t u.
Throw:<bin1 mor word past act first plur>
        = r a m a j n a:.
Throw:<bin1 mor word past act secnd sing
        masc> = r a m a j t a.
Throw:<bin1 mor word past act secnd sing
        femn> = r a m a j t i.
Throw:<bin1 mor word past act secnd plur
        masc> = r a m a j t u m.
```

```
Throw:<bin1 mor word past act secnd plur
       femn> = r a m a j t u n n a.
Throw:<bin1 mor word past act third sing
       masc> = r a m a a.
Throw:<bin1 mor word past act third sing
       femn> = r a m a t.
Throw:<bin1 mor word past act third plur
       masc> = r a m a w.
Throw:<bin1 mor word past act third plur
       femn> = r a m a j n a.
Throw:<bin1 mor word pres act first sing>
       = a r m i:.
Throw:<bin1 mor word pres act first plur>
       = n a r m i:.
Throw:<bin1 mor word pres act secnd sing
       masc> = t a r m i:.
Throw:<bin1 mor word pres act secnd sing
       femn> = t a r m i: n a.
Throw:<bin1 mor word pres act secnd plur
       masc> = t a r m u: n a.
Throw:<bin1 mor word pres act secnd plur
       femn> = t a r m i: n a.
Throw:<bin1 mor word pres act third sing
       masc> = j a r m i:.
Throw:<bin1 mor word pres act third sing
       femn> = t a r m i:.
Throw:<bin1 mor word pres act third plur
       masc> = j a r m u: n a.
Throw:<bin1 mor word pres act third plur
       femn> = j a r m i: n a.

Fold:<bin1 mor word past act first sing>
       = T a w a j t u.
Fold:<bin1 mor word past act first plur>
       = T a w a j n a:.
Fold:<bin1 mor word past act secnd sing
       masc> = T a w a j t a.
Fold:<bin1 mor word past act secnd sing
       femn> = T a w a j t i.
Fold:<bin1 mor word past act secnd plur
       masc> = T a w a j t u m.
Fold:<bin1 mor word past act secnd plur
       femn> = T a w a j t u n n a.
Fold:<bin1 mor word past act third sing
       masc> = T a w a:.
Fold:<bin1 mor word past act third sing
       femn> = T a w a t.
Fold:<bin1 mor word past act third plur
       masc> = T a w a w.
Fold:<bin1 mor word past act third plur
       femn> = T a w a j n a.
Fold:<bin1 mor word pres act first sing>
       = a T w i:.
Fold:<bin1 mor word pres act first plur>
       = n a T w i:.
Fold:<bin1 mor word pres act secnd sing
       masc> = t a T w i:.
Fold:<bin1 mor word pres act secnd sing
       femn> = t a T w i: n a.
Fold:<bin1 mor word pres act secnd plur
       masc> = t a T w u: n a.
Fold:<bin1 mor word pres act secnd plur
       femn> = t a T w i: n a.
Fold:<bin1 mor word pres act third sing
       masc> = j a T w i:.
Fold:<bin1 mor word pres act third sing
       femn> = t a T w i:.
Fold:<bin1 mor word pres act third plur
       masc> = j a T u: n a.
Fold:<bin1 mor word pres act third plur
       femn> = j a T w i: n a.
```

Using the Yago ontology as a resource for the enrichment of Named Entities in Arabic WordNet

Lahsen Abouenour[1], Karim Bouzoubaa[1], Paolo Rosso[2]
1 Mohammadia School of Engineers, Med V University
Rabat, Morocco
2 Natural Language Engineering Lab. - ELiRF, Universidad Politécnica
Valencia, Spain
E-mail: abouenour@yahoo.fr, karim.bouzoubaa@emi.ac.ma, prosso@dsic.upv.es

Abstract

The development of sophisticated applications in the field of the Arabic Natural Language Processing (ANLP) depends on the availability of resources. In the context of previous works related to the domain of the Arabic Question/Answering (Q/A) systems, a semantic Query Expansion approach using Arabic WordNet (AWN) has been evaluated. The obtained results, although AWN (one of the rare resources) has a low coverage of the Arabic language, showed that it helps to improve performances. The evaluation process integrates a Passage Retrieval (PR) system which helps to rank the returned passages according to their structure similarity with the question. In this paper, we investigate the usefulness of enriching AWN by means of the Yago ontology. Preliminary experiments show that this technique helps to extend and improve the processed questions.

1. Introduction

Arabic Natural Language Processing (ANLP) has known interesting attempts in the last years especially in morphology and less advanced Information Retrieval (IR) systems. However, the development of more sophisticated applications such as Question/Answering (Q/A), Search Engines (SEs) and Machine Translation (MT) has still a common problem: the lack of available electronic resources.

The Arabic WordNet (AWN) ontology (Elkateb et al., 2006) is one of these few resources. The AWN[1] is a lexical ontology composed of 23,000 Arabic words and 10 thousands of synsets (sets of words having a common meaning). The design of AWN presents many advantages for its use in the context of ANLP. Indeed, AWN has the same structure as the Princeton WordNet (PWN) (Fellbaum, 2000) and WordNets of other languages. The AWN ontology is also a semantic resource since it contains relations between its synsets and links to the concepts of the Suggested Upper Model Ontology (SUMO) (Niles & Pease, 2003). The advantages described above show that AWN can contribute in the development of sophisticated applications as well as the development of cross-language systems.

In a previous work on Arabic Q/A, (Abouenour et al., 2009b) proposed a Query Expansion (QE) approach which relies on the AWN content and its semantic relations, namely synonymy, hypernymy, hyponymy and definition. The proposed approach has improved the performances in terms of accuracy, the MRR[2] and the number of answered questions.

The reached performances have been considered encouraging for the following reasons:

- a number of 2,264 of well-known question sets in the field of Q/A and IR, namely the TREC[3] and CLEF[4] collections, were used;
- The difference of performances before and after using AWN is significant;
- Experiments have been conducted in an open domain (the web) which is a challenging context for Q/A systems.

Even though AWN has a low coverage of the Arabic language regarding other languages such as English, it helped to improve performances.

In order to enhance further performances, the idea is to develop and use a more enriched release of the AWN ontology. The enrichment of AWN could be done according to different lines: adding new synsets, enriching the existing synsets, enriching the hypoymy/hypernymy relations, verb categorization, Named Entity (NE) synsets, gloss, forms (for instance broken plurals), etc. The aim was focusing on the AWN lacks related to the used question collections. Therefore, the authors have performed an analysis of

[1] http://www.globalwordnet.org/AWN/

[2] Mean Reciprocal Rank (MRR) is defined as the average of the reciprocal ranks of the results for a sample of queries (the reciprocal rank of a query response is the multiplicative inverse of the rank of the correct answer).

[3] Text REtrieval Conference, http://trec.nist.gov/data/qa.html

[4] Cross Language Evaluation Forum, http://www.clef-campaign.org

the questions which contain keywords that can not be found in AWN (not extensible questions) and those for which the system could not reach the expected answer (not answered questions). For the two types of questions, they investigated either the keywords forming the questions and the type of the expected answer.

The analysis showed that for a high percentage of the considered questions, both the question keywords and answers are NEs. Hence, the enrichment of the NE content in the AWN ontology could help us to reach higher performances.

In this paper, we present an attempt to perform an automatic AWN enrichment for the NE synsets. Indeed, the use of a NER system (if such system is available and accurate in the context of the Arabic language) allows only identifying NE and information related to them whereas adding NE in AWN helps also to identify synsets which are semantically related to them (synonyms, subtypes, supertypes, etc.). Moreover, such enrichment could be also useful in the context of other ANLP and Cross-language tasks.

The current work is based on the Yago[5] ontology which contains 2 million entities (such as persons, organizations, cities, etc.). This ontology (Suchanek et al., 2007) contains 20 million facts about these entities. The main reasons behind using this resource are:

- its large coverage of NEs can help to improve performances in the context of Arabic Q/A systems;
- its connection to the PWN and the SUMO ontology (Gerard et al., 2008) can help us to transfer the large coverage of Yago to the AWN ontology.

The rest of the paper is structured as follows: Section 2 describes works using AWN; Section 3 presents the technique proposed for the AWN enrichment; Section 4 is devoted to the presentation of the preliminary experiments that we have conducted on the basis of the Yago content; in Section 5 we draw the main conclusions of this work and we discuss future work.

2. Arabic WordNet in previous works

There are many works that have integrated AWN as a lexical or a semantic resource. To our knowledge, most of these works belongs to the Arabic IR and Q/A fields. Indeed, in (El Amine, 2009), AWN has been used as a lexical resource for a QE process in the context of the IR task.

In the context of Q/A systems, authors in (Brini et al., 2009) have proposed an Arabic Q/A system called QASAL. They have reported that it will be necessary in future works to consider, the synonymy relations between AWN synsets at the question analysis stage of the proposed system. In (Benajiba et al., 2009), the authors have reported that the use of AWN would allow exploring the impact of semantic features for the Arabic Named Entity Recognition (NER) task which is generally included in the first question analysis step of a Q/A process (generally composed by three steps: question analysis, passages re-trieval and answer extraction).

In (Abouenour et al., 2008; Abouenour et al., 2009a), the authors have shown how it is possible to build an ontology for QE and semantic reasoning in the context of the Arabic Q/A task. In addition, the usefulness of AWN as a semantic resource for QE has been proved in the recent work of (Abouenour et al., 2009a) where the authors have considered not only the lexical side of AWN, but also its semantic and knowledge parts. Moreover, the QE process based on AWN has been used together with a structure-based technique for Passage Retrieval (PR). Indeed, the first step of our approach is retrieving a large number of passages which could contain the answer to the entered question. Generally, the answer is expected to appear in those passages nearer to the other keywords of the question or to the terms which are semantically related to those keywords. Therefore, new queries from the question were generated by replacing a keyword by its related terms in AWN regarding the four semantic relations mentioned previously.

In the second step of the described approach, the returned passages have to be ranked according to the structure similarity between the passages and the question. Thus, this step allows decreasing the number of passages to be processed at the answer extraction stage.

The conducted experiments showed an improvement of performances thanks to our two steps approach based on the AWN ontology for QE and the Java Information Retrieval System[6] (JIRS) (Gomez et al., 2007) for structure based PR. The analysis of the obtained results showed that:

- A high percentage (46.2%) of the TREC and CLEF questions are of NEs;
- The enrichment of the NE content in AWN will allow extending 69% of the non extensible questions;
- For a high percentage of the considered questions (50%), we can reach a similarity (between the question and passages) equal or higher than 0.9 and an average of 0.95 (max is 1) by using AWN together with JIRS.

[5] Yet an Other Great Ontology, available at http://www.mpi-inf.mpg.de/yago-naga/yago/downloads.html

[6] http://jirs.dsic.upv.es

Thus, according to this analysis, the priority in terms of AWN enrichment is clear: in order to evaluate the QE and structure-based approach, we have to enlarge and refine the coverage, hierarchy and relations related to the NE synsets in AWN.

In the next section, we describe how resources belonging to other languages could be used for the enrichment of the NE content in AWN.

3. Enrichment of Arabic WordNet using Yago

According to the great number of words of the Modern Standard Arabic (MSA) language, the current release of AWN which has been manually built has still to be enlarged. The automatic enrichment is a promising way to reach a large coverage by AWN regarding the MSA. In this context, authors in (Al Khalifa and Rodriguez, 2009) have proposed a new approach for extending automatically the NE coverage of AWN. This approach relies on Wikipedia[7]. The evaluation done in that work shows that 93.3% of the NE synsets which was automatically recovered are correct. However, due to the small size of the Arabic wikipedia, only 3,854 Arabic NEs have been recovered.

Our approach proposes using a freely available ontology with a large coverage of NE instead of the Arabic Wikipedia. In addition to Yago, the field of open source ontologies provides interesting resources and attempts which belong either to the specific and open domain category: OpenCyc (Matuszek et al., 2006), Know-ItAll (Etzioni et al., 2004), HowNet[8], SNOMED[9], GeneOntology[10], etc.

For the purpose of the current work, we have been interested in using Yago for the following reasons (Suchanek et al., 2007):

- It covers a great amount of individuals (2 millions NEs),
- It has a near-human accuracy around 95%,
- It is built from WordNet and Wikipedia,
- It is connected with the SUMO ontology,
- It exists in many formats (XML, SQL, RDF, Notation 3, etc.) and is available with tools[11] which facilitate exporting and querying it.

The Yago ontology contains two types of information: entities and facts. The former are NE instances (from Wikipedia) and concepts (from WordNet), whereas the latter are facts which set a relation between these entities. To our knowledge Yago has been used as a semantic resource in the context of IR systems (Pound et al., 2009).

As we are interested in enriching the NE content of AWN, a translation stage has to be considered in our process. In (Al Khalifa and Rodriguez, 2009), authors used the Arabic counterpart of the English Wikipedia pages as a translation technique. In the current work, we consider instead the Google Translation API[12] (GTA) because its coverage for NEs written in Arabic is higher than the one of Arabic Wikipedia. In addition, translating a word using GTA is faster. Indeed, the result of a translation using Arabic Wikipedia needs to be disambiguated as many possible words are returned. This is not the case for the GTA.

The enrichment concerns both adding new individuals (NE) and adding their supertypes. These supertypes are very important and useful in our QE process combined to the structure-based PR system (JIRS). In order to show this usefulness, let us consider the example of the TREC question " متى ولد ليندون جونسون ؟" (When was Lindon Johnson born?). When we query a search engine using this question, the two following passages could be returned:

سنة 1908 و هو العام الذي **ولد** فيه ***ليندون جونسون*** ...	The year 1908 which is the year of birth of Lindon Johnson ...
ولد الرئيس الأمريكي ليندون جونسون يوم 27 أغسطس 1908 ...	The American president Lindon Johnson was born in 27 August 1908 ...

According to the two passages above, the JIRS system will consider the first passage as being the most relevant. Indeed, since the two passages contain the keywords of the question (ولد، ليندون جونسون), the similarity of the structure of each passage to the one of the question is the criterion to be used to compare them. The second passage contains a structure similar to the question with two additional terms (which are not among the question keywords) whereas in the first passage only one additional term appears (fyh - فيه). Therefore, the latter is considered more similar to the question than the former one. After enriching AWN by the NE ليندون جونسون and its supertypes such as رئيس أمريكي (r}ys >mryky : US President), we can consider, in the query processed by JIRS, the extended form of the question where the NE is preceded by its supertype الرئيس الأمريكي. In this case, the two terms الرئيس and الأمريكي are considered as being among the question keywords. Hence, the structure of the second passage would then be considered by JIRS as the most similar to the structure of the question. The second passage is the one containing the expected answer in a structure which

[7] www.wikipedia.org/
[8] www.keenage.com/html/e_index.html
[9] www.snomed.org
[10] www.geneontology.org
[11] http://www.mpi-inf.mpg.de/yago-naga/yago/downloads.html
[12] http://code.google.com/p/google-api-translate-java/

structure which can be easy to process by the answer extraction module. In order to enrich the NE content in AWN, we have adopted an approach composed of seven steps. Figure 1 below illustrates these steps.

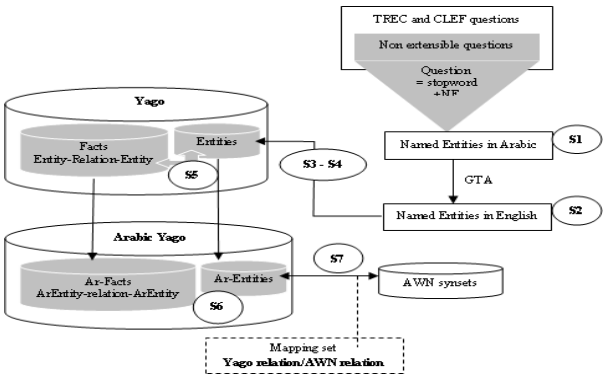

Figure 1: Steps of our approach for the enrichment of the NE content in AWN

As we can see, for the purpose of the current paper, we are interested in the enrichment of the NE part of AWN for the not extensible questions (547 TREC and CLEF questions). In order to do so, our approach relies on the following steps:

(S1) For each considered question, we extract the NE in Arabic;
(S2) Using the GTA, we translate the NE into English (GTA performs a disambiguation process);
(S3) We extract the Yago Entity related to the NE translated into English;
(S4) We extract the Yago Facts related to the Yago Entity extracted in the previous step;
(S5) In this step, we have a sub release of Yago related to the considered questions;
(S6) Using the GTA, we translate the content (entities and facts) of the sub release of Yago built in step five;
(S7) We perform a mapping between the NEs contained in the Arabic Yago (of step S4) and their related entries in AWN according to synonymy, hypernymy, hyponymy and SUMO relations.

After performing these steps, we have an enriched release of AWN which we consider in our new experiments. The obtained results in the enrichment and experimental processes are described in the next section.

4. Preliminary Experiments

As we have mentioned in the previous section, our focus is devoted to the NEs which appear in the not extensible questions. The number of these questions is 547. There are some NEs which appear in many questions. The number of distinct NEs is 472.

After performing steps 3 and 4, 374 distinct NEs (79%) have been identified within the Yago ontology. A number of 59,747 facts concern the identified Yago entities, with an average of 160 facts per entity. The average of the confidence related to these facts around 0.97 (the max is 1). The Yago ontology contains 96 relations. We have identified 43 relations in the facts corresponding to the NEs extracted from the considered questions. The TYPE relation is the first one to be considered in our approach for the enrichment of NEs in the AWN. For the purpose of the current work, we have considered only the facts containing a TYPE relation between a Yago entity and a WordNet concept. From the 374 NEs identified in Yago, 204 of them (around 55%) have a TYPE relation with a WordNet concept.

Relying on these relations on one hand and on the relation between the AWN synsets and the WordNet synsets on the other hand, we were able to connect 189 Yago entities (roughly 51% of the NEs of the considered questions) with the corresponding AWN synsets.

In order to connect the rest of NEs (185) with the AWN synsets (102 distinct synsets), we have set, in the context of the step S7 mentioned previously, different mappings between the relations used in the Yago facts and the corresponding AWN synsets. For instance, the second arguments of the relations "**citizenOf**", "**livesIn**", "**bornIn**", "**hasCapital**" or "**locatedIn**" are candidate hyponyms of the AWN synset "**مدينة**" (mdynp : city).

The enriched release of AWN that we have built using Yago helped us extending more questions and conducting preliminary experiments in the same way of (Abouenour et al., 2009a). Table 1 shows the obtained results.

Measures	before Yago	Using Yago
Accuracy	17,49%	23,53%
MRR	7,98	9,59
Number answered questions	23,15%	31,37%

Table 1: Results of preliminary experiments related the non extensible questions.

As we can see, performances in terms of accuracy, MRR and the number of answered questions have been improved after using our semantic QE which relies on the AWN release enriched with Yago.

5. Conclusion and Future Works

In this paper, we have proposed an approach to enrich AWN from the available content of the Yago ontology. The enrichment process was possible thanks to the connection existing between Yago entities and

WordNet on one hand and between WordNet and AWN on the other hand. In the preliminary experiments that we have conducted, we have considered the previous semantic QE approach which relies now on the new content of AWN. These experiments show an improvement in terms of accuracy, MRR and the number of answered questions.

In the current work, we have considered only the relations of Yago which allow a direct mapping between its entities and the AWN synsets. Therefore, considering the other relations and the whole content of Yago is among the intended future works.

Acknowledgement

This research work is the result of the collaboration in the framework of the bilateral Spain-Morocco AECID-PCI C/026728/09 research project. The third author thanks also the TIN2009-13391-C04-03 research project.

References

Al Khalifa M. and Rodríguez H. 2009. "Automatically Extending NE coverage of Arabic WordNet using Wikipedia". In Proc. Of the 3rd International Conference on Arabic Language Processing CITALA2009, Rabat, Morocco, May, 2009.

Abouenour L., Bouzoubaa K., Rosso P., 2009. "Three-level approach for Passage Retrieval in Arabic Question /Answering Systems". In Proc. Of the 3rd International Conference on Arabic Language Processing CITALA2009, Rabat, Morocco, May, 2009.

Abouenour L., Bouzoubaa K., Rosso P., 2009. "Structure-based evaluation of an Arabic semantic Query Expansion using the JIRS Passage Retrieval system" . In: Proc. Workshop on Computational Approaches to Semitic Languages, E-ACL-2009, Athens, Greece.

Abouenour L., Bouzoubaa K., Rosso P. 2008. Improving Q/A Using Arabic Wordnet. In: Proc. *The 2008 International Arab Conference on Information Technology (ACIT'2008),* Tunisia, December.

Benajiba Y., Mona D., Rosso P. Using Language Independent and Language Specific Features to Enhance Arabic Named Entity Recognition. In: IEEE Transactions on Audio, Speech and Language Processing. Special Issue on Processing Morphologically Rich Languages, Vol. 17, No. 5, July 2009.

Brini W., Ellouze M., Hadrich Belguith L. 2009. *QA SAL :* "Un système de question-réponse dédié pour les questions factuelles en langue Arabe*". In: 9ème Journées Scientifiques des Jeunes Chercheurs en Génie Electrique et Informatique, Tunisia.* (in French).

El Amine M. A. 2009. Vers une interface pour l'enrichissement des requêtes en arabe dans un système de recherche d'information. In Proceedings of the 2nd Conférence Internationale sur l'informatique et ses Applications (CIIA'09) Saida, Algeria, May 3-4, 2009.

Elkateb S., Black W., Vossen P., Farwell D., Rodríguez H., Pease A., Alkhalifa M. 2006. "Arabic WordNet and the Challenges of Arabic". *In proceedings of Arabic NLP/MT Conference*, London, U.K.

Etzioni O., M. J. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2004. Web-scale information extraction in KnowItAll. In WWW, 2004.

Fellbaum C. 2000. "WordNet: An Electronic Lexical Database". *MIT Press*, *cogsci.princeton.edu/~wn,* September 7.

Gerard D. M., Suchanek F. M., Pease A. Integrating YAGO into the Suggested Upper Merged Ontology. 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008). Dayton, Ohio, USA (2008).

Gómez J. M., Rosso P., Sanchis E. 2007. Re-ranking of Yahoo snippets with the JIRS Passage Retrieval system. In: Proc. Workshop on Cross Lingual Information Access, CLIA-2007, 20th Int. Joint Conf. on Artificial Intelligence, IJCAI-07, Hyderabad, India, January 6-12.

Matuszek C., J. Cabral, M. Witbrock, and J. De Oliveira. An introduction to the syntax and content of Cyc. In AAAI Spring Symposium, 2006.

Niles I., Pease A. 2003. "Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology." *In Proceedings of the 2003 International Conference on Information and Knowledge Engineering*, Las Vegas, Nevada.

Pound J., Ihab F. I., and Weddell. G. 2009. QUICK: Queries Using Inferred Concepts fromKeywords Technical Report CS-2009-18. Waterlo, Canada.

Rodríguez H., Farwell D., Farreres J., Bertran M., Alkhalifa M., Antonia Martí M., Black W., Elkateb S., Kirk J., Pease A., Vossen P., and Fellbaum C. 2008. Arabic WordNet: Current State and Future Extensions in: Proceedings of the Fourth International GlobalWordNet Conference - GWC 2008, Szeged, Hungary, January 22-25, 2008.

Suchanek, F. M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In Proc. of the 16th WWW, pp. 697-706 (2007).

Light Morphology Processing for Amazighe Language

Fadoua Ataa Allah, Siham Boulaknadel
CEISIC, IRCAM
Avenue Allal El Fassi, Madinat Al Irfane, Rabat, Morocco
E-mail: {ataaallah, boulaknadel}@ircam.ma

Abstract

In the aim to allow the Amazighe language an automatic processing, and integration in the field of Information and Communication Technology, we have opted in the Royal Institute of Amazighe Culture "IRCAM" for an innovative approach of progressive realizations. Thus since 2003, researchers in the Computer Sciences Studies, Information Systems and Communications Center "CEISIC" have paved the way for elaborating linguistic resources, basic natural language processing tools, and other advanced scientific researches by encoding Tifinaghe script and developing typefaces.
In this context, we are trying through this paper to develop a computationally stemming process which is based on analyzing words to their stems. This process consists in splitting Amazighe words into constituent stem part and affix parts without doing complete morphological analysis. This approach of light stemming will conflate word variants into a common stem in order to be used in natural language applications such as indexation, information retrieval systems, and classification.

1. Introduction

Stemming has been widely used in several fields of natural language processing such as data mining, information retrieval, machine translation, document summarisation, and text classification, in which the identification of lexical occurrences of words referring to some central idea or 'meaning' is involved. Indeed, the lexical analysis is mainly based on word occurrences, which require some form of morphological conflation that could range from removing affixes to using morphological word structures.

In literature, many strategies of stemming algorithms have been published for different languages, such as English (Lovins 1968; Porter, 1980), French (Savoy, 1993; Paternostre et al., 2002), and Arabic (Larkey et al., 2002; Taghva et al., 2005; Al-Shammari and Lin, 2008). In general, the stemmer structures vary considerably depending on the morphology of languages. For Indo-European languages, most basic techniques consist on removing suffixes; while, for the Afro-Asiatic ones, these techniques are extended to stripping prefixes.

In practice, affixes may alter the meaning of words. So, the fact to remove them would greatly discard vital information. In the Indo-European languages, prefixes modify the word meaning which make their deletion not helpful. While in the Afro-Asiatic languages, the prefixes are also used to fit the word for its syntactic role. Thus, in this paper, we propose an Amazighe stemming algorithm that consists in removing the common inflectional morphemes placed at the beginning and/or the end of words.

The remaining of the paper is organized as follows: in Section 2, we give a brief description of the Moroccan standard Amazighe language. Then, in Section 3, we give an overview about the Amazighe language characteristics. In Section 4, we present our light stemming algorithm. Finally, section 5 gives general conclusions, and draws some perspectives.

2. Moroccan Standard Amazighe Language

The Amazighe language, known as Berber or Tamazight, is a branch of the Afro-Asiatic (Hamito-Semitic) language family. It covers the Northern part of Africa which extends from the Red Sea to the Canary Isles, and from the Niger in the Sahara to the Mediterranean Sea. In Morocco, this language is divided into three mean regional varieties: Tarifite in North, Tamazight in Central Morocco and South-East, and Tachelhite in the South-West and the High Atlas. Even though 50% of the Moroccan population are amazighe speakers, the Amazighe language was exclusively reserved for familial and informal domains (Boukous, 1995). However, in the last decade, this language has become institutional.

Since the ancient time, the Amazighe language has its own writing that was adapted by the Royal Institute of the Amazighe Culture (IRCAM) in 2003, to provide an adequate and usable standard alphabetic system called Tifinaghe-IRCAM. This system contains:

- 27 consonants including: the labials (ⴱ, ⴼ, ⵎ), dentals (ⵜ, ⵟ, ⴷ, ⴹ, ⵏ, ⵍ, ⵕ, ⵝ), the alveolars (ⵙ, ⵣ, ⵚ, ⵥ), the palatals (ⵛ, ⵊ), the velar (ⴽ, ⵅ), the labiovelars (ⴽⵯ, ⵅⵯ), the uvulars (ⵇ, ⵅ, ⵖ), the pharyngeals (ⵄ, ⵃ) and the laryngeal (ⵀ);
- 2 semi-consonants: ⵢ and ⵡ;
- 4 vowels: three full vowels ⴰ, ⵉ, ⵓ and neutral vowel (or schwa) ⴻ which has a rather special status in amazighe phonology.

Furthermore, the IRCAM has recommended the use of the International symbols for punctuation markers: " " (space), ".", ",", ";", ":", "?", "!", "…"; the standard numeral used in Morocco (0, 1, 2, 3, 4, 5, 6, 7, 8, 9); and the horizontal direction from left to right for Tifinaghe writing (Ameur et al., 2004).

3. Amazighe Language Characteristics

The purpose of this section is to give an overview of the morphological properties of the main syntactic amazighe categories, which are the noun, the verb, and the particles (Boukhris et al., 2008; Ameur et al., 2004).

3.1 Noun

In Amazighe language, noun is a lexical unit, formed from a root and a pattern. It could occur in a simple form (ⴰⵔⴳⴰⵣ "argaz" *the man*), compound form (ⵓⵖⴼⵓⵖ "buhyyuf" *the famine*), or derived one (ⴰⵎⵙⴰⵡⴰⴹ "amsawaḍ " *the communication*). This unit varies in gender, number and case.

- Gender: Nouns are categorised by grammatical gender: masculine or feminine. Generally, the masculine begin with an initial vowel ⴰ "a", ⵉ "i", or ⵓ "u". While, the feminine, used also to form diminutives and singulatives, is marked with the circumfix ⵜ…ⵜ "t…t" (ⴰⵎⵔⴰⴷⴰⵇ "amḥ ḍ aṛ "masc., ⵜⴰⵎⵔⴰⴷⴰⵇⵜ "tamḥ ḍ aṛ t" fem. *the student*).
- Number: There are two types: singular and plural, which has three forms. The external plural consists in changing the initial vowel, and adding the suffix ⵏ or one of its variants ⵉⵏ "in", ⴰⵏ "an", ⵢⵏ "yn", ⵡⵏ "wn", ⴰⵡⵏ "awn", ⵉⵡⵏ "iwn", ⵜⵏ "tn" (ⵉⵎⵔⴰⴷⴰⵇⵏ "im amḥ ḍ arn" masc., ⵜⵉⵎⵔⴰⴷⴰⵇⵉⵏ "timḥ ḍ arin" fem. *students*). The broken plural involves a change in the vowels of the noun (ⴰⴷⵔⴰⵔ "adrar" *mountain* → ⵉⴷⵓⵔⴰⵔ idurar *mountains*, ⵜⵉⵖⵎⵙⵜ "tiγmst" *tooth* → ⵜⵉⵖⵎⴰⵙ "tiγmas" *teeth*). The mixed plural is formed by the combination of vowels' change and the use, sometimes of the suffixation ⵏ (ⵉⵣⵉ "izi" *fly* → ⵉⵣⴰⵏ "izan" *flies*, ⴰⵎⵖⵖⵓⵔⵓ "amgguru" *last* → ⵉⵎⵖⵖⵓⵔⴰ "imggura" *lasts*).
- Case: Two cases are distinguished. The free case is unmarked, while the construct one involves a variation of the initial vowel (ⴰⵔⴳⴰⵣ "argaz" *man* → ⵓⵔⴳⴰⵣ, "urgaz" ⵜⴰⵎⵖⴰⵔⵜ "tamγart" *woman* → ⵜⵎⵖⴰⵔⵜ "tmγart").

3.2 Verb

The verb, in Amazighe, has two forms: basic and derived forms. The basic form is composed of a root and a radical (ⴼⴼⵖ "ffγ" *leave*), while the derived one is based on the combination of a basic form and one of the following prefixes morphemes: ⵙ/ⵙⵙ "s/ss", ⵜⵜ "tt" and ⵎ/ⵎⵎ "m/mm" (ⵙⵙⵓⴼⵖ "ssufγ" *bring out*). Whether basic or derived, the verb is conjugated in four aspects: aorist, imperfective, perfect, and negative perfect. Moreover, it is constructed using the same personal markers for each mood, as represented in Table1.

3.3 Particles

In Amazighe language, particle is a function word that is not assignable to noun neither to verb. It contains pronouns; conjunctions; prepositions; aspectual, orientation and negative particles; adverbs; and subordinates. Generally, particles are uninflected words. However in Amazighe, some of these particles are flectional, such as the possessive and demonstrative pronouns (ⵜⴰ "ta" *this* (fem.) → ⵜⵉⵏⴰ "tina" *these* (fem.)).

4. Light Stemming Algorithm

The light stemming refers to a process of stripping off a small set of prefixes and/or suffixes, without trying to deal with infixes, or recognizing patterns and finding roots (Larkey, 2002). As a first edition of such work in the IRCAM, with regard to the lack of huge digital corpus availability, our method is based only on the composition of words that is usually formed in the Moroccan standard Amazighe language as a sequence of prefix, core, and suffix. We are assuming that we are not making use of any stem dictionary or exception list. Our algorithm is merely based on an explicit list of prefixes and suffixes that need to be stripped in a certain order. This list is derived from the common inflectional morphemes of gender, number and case for nouns; personal markers, aspect and mood for verbs; and affix pronouns for kinship nouns and prepositions. While, the derivational morphemes are not included in order to keep the semantic meaning of words. It is very reasonable to conflate the noun ⵜⴰⵓⴱⴰⵜ "tarbat" *girl* with its masculine form "ⴰⵓⴱⴰ" arba *boy*; while it seems unreasonable, for some application like information retrieval, to conflate the derived verb ⵙⵙⵓⴼⵖ "ssufγ" *bring out* with the simple form ⴼⴼⵖ "ffγ" *leave*.

The set of prefixes and suffixes, that we have identified, are classified to five groups ranged from one character to five characters.

4.1 Prefix Set

- One-character: ⴰ, ⵉ, ⵏ, ⵓ, ⵜ.
- Two-character: ⵏⴰ, ⵏⵉ, ⵏⵓ, ⵜⴰ, ⵜⵉ, ⵜⵓ, ⵜⵜ, ⵡⴰ, ⵡⵓ, ⵢⴰ, ⵢⵉ, ⵢⵓ.
- Three-character: ⵉⵜⵜ, ⵏⵜⵜ, ⵜⵜⴰ, ⵜⵜⵉ.
- Four-character: ⵉⵜⵜⴰ, ⵉⵜⵜⵉ, ⵏⵜⵜⴰ, ⵏⵜⵜⵉ, ⵜⵓⵜⵜ.
- Five-character: ⵜⵓⵜⵜⴰ, ⵜⵓⵜⵜⵉ.

4.2 Suffix Set

- One-character: ⴰ, ⴷ, ⵉ, ⵇ, ⵎ, ⵏ, ⵢ, ⵙ, ⵜ.
- Two-character: ⴰⵏ, ⴰⵜ, ⵉⴷ, ⵉⵎ, ⵉⵏ, ⵉⵢ, ⵎⵜ, ⵏⵜ, ⵓⵏ, ⵙⵏ, ⵙⵜ, ⵢⵏ.
- Three-character: ⴰⵎⵜ, ⴰⵏⵜ, ⴰⵡⵏ, ⵉⵎⵜ, ⵉⵏⵜ, ⵉⵡⵏ, ⵓⵏⵜ, ⵓⵙⵏ, ⵓⵙⵜ, ⵜⵉⵏ, ⵜⵓⵏ, ⵜⵓⵜ.
- Four-character: ⵜⵓⵏⵜ, ⵜⵓⵙⵏ.

	Indicative mood			Imperative mood			Participial mood
		Masculine	Feminine		Masculine	Feminine	Masculine / Feminine
Singular	1st pers. 2nd pers. 3rd pers.	... ⵖ ⵜ ... ⴰ ⵙ ... _____	... ⵖ ⵜ ... ⴰ ⵜ ... _____	2nd pers.	... Ø	… Ø	ⵙ….ⵏ
Plural	1st pers. 2nd pers. 3rd pers.	ⵏ ... ⵜ ... ⵛ ... ⵏ	ⵏ ... ⵜ ... ⵛⵜ ... ⵏⵜ	2nd pers.	... ⵕⵜ/ⵜ/ⵛ	... ⵕⵛⵜ/ ⵛⵜ	….ⵏⵙⵏ

Table 1: Personal markers for the indicative, imperative and participial moods

Based on this list of affixes and on theoretical analysis, we notice that the proposed amazighe light stemmer could make two kinds of errors:

- The understemming errors, in which words referring to the same concept are not reduced to the same stem, such the case of the verb ⵅⵅⵖ "ffγ" *leave* that ends with the character ⵖ "γ", which coincides with the 1st singular personal marker. So, the stem ⵅⵅⵖ "ffγ" of the verb when is conjugated in the perfect aspect for the 1st singular person ⵅⵅⵖⵖ "ffγγ" *I left* will not be conflated with stem ⵅⵅ "ff" of the 3rd singular masculine person ⵙⵅⵅⵖ "iffγ" *he left*.
- The overstemming errors, in which words are converted to the same stem even though they refer to distinct concepts, such the example of the verb ⵅ "g" *do* and the noun ⴰⵅⴰ "aga" *bucket*. The stem ⵅ "g" of the verb when is conjugated in the perfect aspect for the 3rd singular masculine person ⵙⵅⴰ "iga" *he did* will be conflated with stem ⵅ "g" of the noun ⴰⵅⴰ "aga".

In general, light stemmers avoid the overstemming errors, especially for the Indo-European languages; however, it is not the case of the Amazighe language. This proves that the Amazighe language constitutes a significant challenge for natural language processing.

5. Conclusion

Stemming is an important technique for highly inflected language such as Amazighe. In this work, we have investigated on the Amazighe language characteristics, and have presented a light stemming approach for Amazighe. We should note that the proposed stemming algorithm is primarily for handling inflections – it does not handle derivational suffixes, for which one would need a proper morphological analyzer.

In attempt to improve the amazighe light stemmer, we plan to build a stem dictionary, to elaborate a set of linguistic rules, and to set a list of exceptions to further extend the stemmer.

6. Appendix

Tifinaghe	Latin Correspondence	Tifinaghe	Latin Correspondence
ⴰ	a	ⵍ	l
ⴱ	b	ⵎ	m
ⴳ	g	ⵏ	n
ⴳⵯ	g^w	ⵓ	u
ⴷ	d	ⵔ	r
ⴹ	ḍ	ⵕ	ṛ
ⴻ	e	ⵖ	γ
ⴼ	f	ⵙ	s
ⴽ	k	ⵚ	ṣ
ⴽⵯ	k^w	ⵛ	c
ⵀ	h	ⵜ	t
ⵃ	ḥ	ⵟ	ṭ
ⵄ	ɛ	ⵡ	w
ⵅ	x	ⵢ	y
ⵇ	q	ⵣ	z
ⵉ	i	ⵥ	ẓ
ⵊ	j		

Table 2: Tifinaghe-Ircam Alphabet

7. References

Al-shammari, E. T., Lin, J. (2008). Towards an error-free Arabic stemming. *Actes de the 2nd ACM workshop on improving non English web searching.* pp.9--16.

Ameur, M., Bouhjar, A., Boukhris, F., Boukouss, A., Boumalk, A., Elmedlaoui, M., Iazzi, E. M., Souifi, H. (2004). *Initiation à la langue amazighe*. Rabat: IRCAM.

Boukhris, F., Boumalk, A., Elmoujahid, E., Souifi, H. (2008). *La nouvelle grammaire de l'amazighe*. Rabat: IRCAM.

Larkey, L. S., Ballesteros, L., Connell, M. (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Cooccurrence Analysis. *In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval*. Tampere, Finland, pp. 275--282.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1), pp. 22--31.

Paternostre, M., Francq, P., Lamoral, J., Wartel, D., Saerens, M. (2002). Carry, un algorithme de désuffixation pour le français. Rapport technique du projet Galilei.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, 14(3), pp.130--137.

Savoy, J. (1993). Stemming of French words based on grammatical categories. *Journal of the American Society for Information Science*, 44(1), pp.1--9.

Taghva, K., Elkhoury, R., Coombs, J. (2005). Arabic stemming without a root dictionary. *In Proceeding of Information Technology: Coding and Computing*. Las Vegas, pp.152--157.

Using Mechanical Turk to Create a Corpus of Arabic Summaries

Mahmoud El-Haj, Udo Kruschwitz, Chris Fox

School of Computer Science and Electronic Engineering
University of Essex
Colchester, CO4 3SQ
United Kingdom
{melhaj,udo,foxcj}@essex.ac.uk

Abstract

This paper describes the creation of a human-generated corpus of extractive Arabic summaries of a selection of Wikipedia and Arabic newspaper articles using Mechanical Turk—an online workforce. The purpose of this exercise was two-fold. First, it addresses a shortage of relevant data for Arabic natural language processing. Second, it demonstrates the application of Mechanical Turk to the problem of creating natural language resources. The paper also reports on a number of evaluations we have performed to compare the collected summaries against results obtained from a variety of automatic summarisation systems.

1. Motivation

The volume of information available on the Web is increasing rapidly. The need for systems that can automatically summarise documents is becoming ever more desirable. For this reason, text summarisation has quickly grown into a major research area as illustrated by the Text Analysis Conference (TAC) and the Document Understanding Conference (DUC) series.

We are interested in the automatic summarisation of Arabic documents. Research in Arabic is receiving growing attention but it has widely been acknowledged that apart from a few notable exceptions—such as the Arabic Penn Treebank[1] and the Prague Arabic Dependency Treebank[2]—there are few publicly available tools and resources for Arabic NLP, such as Arabic corpora, lexicons and machine-readable dictionaries, resources that are common in other languages (Diab et al., 2007) although this has started to change in recent years (Maegaard et al., 2008; Alghamdi et al., 2009). Some reasons for this lack of resources may be due to the complex morphology, the absence of diacritics (vowels) in written text and the fact that Arabic does not use capitalisation. Tools and resources however are essential to advance research in Arabic NLP. In the case of summarisation tasks, most of the activity is concerned with the English language—as with TAC and DUC. This focus is reflected in the availability of resources: in particular, there is no readily available "gold standard" for evaluating Arabic summarisers.

Tools and resources are essential to advance research in Arabic NLP, but generating them with traditional techniques is both costly and time-consuming. It is for this reason that we considered using Amazon's Mechanical Turk[3]—an online marketplace for work that requires human intelligence—to generate our own reference standard for extractive summaries.

2. Related Work

There are various approaches to text summarisation, some of which have been around for more than 50 years (Luhn, 1958). These approaches include single-document and multi-document summarisation. One of the techniques of single-document summarisation is summarisation through extraction. This relies on the idea of extracting what appear to be the most important or significant units of information from a document and then combining these units to generate a summary. The extracted units differ from one system to another. Most of the systems use sentences as units while others work with larger units such as paragraphs.

Evaluating the quality and consistency of a generated summary has proven to be a difficult problem (Fiszman et al., 2009). This is mainly because there is no obvious ideal summary. The use of various models for system evaluation may help in solving this problem. Automatic evaluation metrics such as ROUGE (Lin, 2004) and BLEU (Papineni et al., 2001) have been shown to correlate well with human evaluations for content match in text summarisation and machine translation (Liu and Liu, 2008; Hobson et al., 2007, for example). Other commonly used evaluations include measuring information by testing readers' understanding of automatically generated summaries.

This very brief review of related work should serve as a motivation for the corpus of Arabic summaries that we have produced for the Arabic NLP community. Our decision to use the Mechanical Turk platform is justified by the fact that it has already been shown to be effective for a variety of NLP tasks achieving expert quality (Snow et al., 2008; Callison-Burch, 2009, for example).

3. The Document Collection

The document collection used in the development of the resource was extracted from the Arabic language version of Wikipedia[4] and two Arabic newspapers; Alrai[5] from Jordan and Alwatan[6] from Saudi Arabia. These sources were chosen for the following reasons.

[1] http://www.ircs.upenn.edu/arabic/
[2] http://ufal.mff.cuni.cz/padt/PADT_1.0/
[3] http://www.mturk.com
[4] http://www.wikipedia.com
[5] http://www.alrai.com
[6] http://www.alwatan.com.sa

1. They contain real text as would be written and used by native speakers of Arabic.

2. They are written by many authors from different backgrounds.

3. They cover a range of topics from different subject areas (such as politics, economics, and sports), each with a credible amount of data.

The Wikipedia documents were selected by asking a group of students to search the Wikipedia website for arbitrary topics of their choice within given subject areas. The subject areas were: art and music; the environment; politics; sports; health; finance and insurance; science and technology; tourism; religion; and education. To obtain a more uniform distribution of articles across topics, the collection was then supplemented with newspaper articles that were retrieved from a bespoke information retrieval system using the same queries as were used for selecting the Wikipedia articles. Each document contains on average 380 words.

4. The Human-Generated Summaries

The corpus of extractive document summaries was generated using Mechanical Turk. The documents were published as "Human Intelligence Tasks" (HITS). The assessors (workers) were asked to read and summarise a given article (one article per task) by selecting what they considered to be the most significant sentences that should make up the extractive summary. They were required to select no more than half of the sentences in the article. Using this method, five summaries were created for each article in the collection. Each of the summaries for a given article were generated by different workers.

In order to verify that the workers were properly engaged with the articles, and provide a measure of quality assurance, each worker was asked to provide up to three keywords as an indicator that they read the article and did not select random sentences. In some cases where a worker appeared to select random sentences, the summary is still considered as part of the corpus to avoid the risk of subjective bias.

The primary output of this project is this corpus of 765 human-generated summaries that we obtained, which is now available to the community.[7] To set the results in context, and illustrate its use, we also conducted a number of evaluations.

5. Evaluations

To illustrate the use of the human-generated summaries from Mechanical Turk in the evaluation of automatic summarisation, we created extractive summaries of the same set of documents using a number of systems, namely:

Sakhr: an online Arabic summariser.[8]

AQBTSS: a query-based document summariser based on the vector space model that takes an Arabic document and a query (in this case the document's title) and returns an extractive summary (El-Haj and Hammo, 2008; El-Haj et al., 2009).

Gen-Summ: similar to *AQBTSS* except that the query is replaced by the document's first sentence.

LSA-Summ: similar to *Gen-Summ*, but where the vector space is tranformed and reduced by applying Latent Semantic Analysis (LSA) to both document and query (Dumais et al., 1988).

Baseline-1: the first sentence of a document.

The justification for selecting the first sentence in *Baseline-1* is the believe that in Wikipedia and news articles the first sentence tends to contain information about the content of the entire article, and is often included in extractive summaries generated by more sophisticated approaches (Baxendale, 1958; Yeh et al., 2008; Fattah and Ren, 2008; Katragadda et al., 2009).

When using Mechanical Turk on other NLP tasks, it has been shown that aggregation of multiple independent annotations from non-experts can approximate expert judgement (Snow et al., 2008; Callison-Burch, 2009; Albakour et al., 2010, for example). For this reason, we evaluated the results of the systems not with the raw results of Mechanical Turk, but with derived *gold standard* summaries, generated by further processing and analysis of the human generated summaries.

The aggregation of the summaries can be done in a number of ways. To obtain a better understanding of the impact of the aggregation method on the results of the evaluation, we constructed three different gold standard summaries for each document. First of all we selected all those sentences identified by at least three of the five annotators (we call this *Level 3* summary). We also created a similar summary which includes all sentences that have been identified by at least two annotators (called *Level 2*). Finally, each document has a third summary that contains all sentences identified by any of the annotators for this document (called *All*). This last kind of summary will typically contain outlier sentences. For this reason, only the first two kinds of aggregated summaries (*Level 2* and *Level 3*) should really be viewed as providing genuine gold standards. The third one (*All*) is considered here just for the purposes of providing a comparison.

A variety of evaluation methods have been developed for summarisation systems. As we are concerned with *extractive* summaries, we will concentrate on results obtained from applying Dice's coefficient (Manning and Schütze, 1999), although we will discuss briefly results from N-gram and substring-based methods ROUGE (Lin, 2004) and AutoSummENG (Giannakopoulos et al., 2008).

5.1. Dice's Coefficient

We used Dice's coefficient to judge the similarity of the sentence selections in the gold-standard extractive summaries — derived from the human-generated, Mechanical Turk summaries — with those generated by *Sakhr*, *AQBTSS*, *Gen-Summ*, *LSA-Summ* and *Baseline-1* (Table 1). Statistically significant differences can be observed in a number

[7] http://privatewww.essex.ac.uk/~melhaj/easc.htm

[8] http://www.sakhr.com

	Sakhr	AQBTSS	Gen-Summ	LSA-Summ	Baseline-1
All	39.07%	32.80%	39.51%	39.23%	25.34%
Level 2	48.49%	39.90%	48.95%	50.09%	26.84%
Level 3	43.40%	38.86%	43.39%	42.67%	40.86%

Table 1: Dice results: systems versus MTurk-derived gold standards.

	Sakhr	AQBTSS	LSA-Summ	Gen-Summ	Baseline-1
Sakhr	—	51.09%	58.77%	58.82%	38.11%
AQBTSS	51.09%	—	54.61%	58.48%	47.86%
LSA-Summ	58.77%	54.61%	—	84.70%	34.66%
Gen-Summ	58.82%	58.48%	84.70%	—	34.99%

Table 2: Dice results: comparing systems.

of cases, but we will concentrate on some more general observations.

We observe that the commercial system *Sakhr* as well as the systems that build a summary around the first sentence most closely approximate the gold standards, i.e. *Level 2* and *Level 3*. This is perhaps not surprising as the overlap with the document's first sentence has been shown to be a significant feature in many summarisers (Yeh et al., 2008; Fattah and Ren, 2008).

It is interesting to note that summaries consisting of a single sentence only (i.e. *Baseline-1*) do not score particularly well. That suggests that the first sentence is important but not sufficient for a good summary. When comparing *Baseline-1* with the *Level 2* and *Level 3* summaries, respectively, we also note how the "wisdom of the crowd" seems to converge on the first sentence as a core part of the summary.

Finally, the system that most closely approximates our *Level 2* gold standard uses LSA, a method shown to work effectively in various NLP and IR tasks including summarisation, e.g. (Steinberger and Ježek, 2004; Gong and Liu, 2001).

We also compared the baseline systems with each other (Table 2). This is to get an idea of how closely the summaries each of these systems produce correlate with each other. The results suggest that the system that extracts the first sentence only does not correlate well with any of the other systems. At the same time we observe that *Gen-Summ* and *LSA-Summ* generate summaries that are highly correlated. This explains the close similarity when comparing each of these systems against the gold standards (see Table 1). It also demonstrates (not surprisingly) that the difference between a standard vector space approach and LSA is not great for the relatively short documents in a collection of limited size.

5.2. Other Evaluation Methods

In addition to using Dice's coefficient, we also applied the ROUGE (Lin, 2004) and AutoSummENG (Giannakopoulos et al., 2008) evaluation methods.

In our experiments with AutoSummENG we obtained values for "CharGraphValue" in the range 0.516–0.586. This indicates how much the graph representation of a model summary overlaps with a given peer summary, taking into account how many times two N-grams are found to be neighbours. *Gen-Summ* and *LSA-Summ* gave the highest values indicating that they produce results more similar to our gold standard summaries than what Sakhr and *AQBTSS* produced.

When applying ROUGE we considered the results of ROUGE-2, ROUGE-L, ROUGE-W, and ROUGE-S which have been shown to work well in single document summarisation tasks (Lin, 2004). In line with the results discussed above, *LSA-Summ* and *Gen-Summ* performed better on average than the other systems in terms of recall, precision and F-measure (when using *Level 2* and *Level 3* summaries as our gold standards). Regarding the other systems, they all performed better than *Baseline-1*.

These results should only be taken to be indicative. Dice's coefficient appears to be a better method for *extractive* summaries as we are comparing summaries on the *sentence* level. It is however worth noting that the main results obtained from Dice's coefficient are in line with results from ROUGE and AutoSummENG.

6. Conclusions and Future Work

We have demonstrated how gold-standard summaries can be extracted using the "wisdom of the crowd".

Using Mechanical Turk has allowed us to produce a resource for evaluating Arabic extractive summarisation techniques at relatively low cost. This resource is now available to the community. It will provide a useful benchmark for those developing Arabic summarisation tools. The aim of the work described here was to create a relatively small but usable resource. We provided some comparison with alternative summarisation systems for Arabic. We have deliberately made no attempt in judging the individual quality of each system. How this resource will be used and how effective it can be applied remains the task of the users of this corpus.

7. References

M-D. Albakour, U. Kruschwitz, and S. Lucas. 2010. Sentence-level attachment prediction. In *Proceedings of the 1st Information Retrieval Facility Conference*, Lecture Notes in Computer Science 6107, Vienna. Springer.

M. Alghamdi, M. Chafic, and M. Mohamed. 2009. Arabic language resources and tools for speech and natural lan-

guage: Kacst and balamand. In *2nd International Conference on Arabic Language Resources & Tools*, Cairo, Egypt.

P. B. Baxendale. 1958. Machine-made index for technical literature—an experiment. *IBM Journal of Research and Development*, 2.

C. Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 286–295. Association for Computational Linguistics.

M. Diab, K. Hacioglu, and D. Jurafsky. 2007. Automatic Processing of Modern Standard Arabic Text. In A. Soudi, A. van den Bosch, and G. Neumann, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, Text, Speech and Language Technology, pages 159–179. Springer Netherlands.

S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. 1988. Using latent semantic analysis to improve access to textual information. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285. ACM.

M. El-Haj and B. Hammo. 2008. Evaluation of query-based Arabic text summarization system. In *Proceeding of the IEEE International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE'08*, pages 1–7, Beijing, China. IEEE Computer Society.

M. El-Haj, U. Kruschwitz, and C. Fox. 2009. Experimenting with Automatic Text Summarization for Arabic. In *Proceedings of the 4^{th} Language and Technology Conference (LTC'09)*, pages 365–369, Poznań, Poland.

M.A. Fattah and Fuji Ren. 2008. Automatic text summarization. In *Proceedings of World Academy of Science*, volume 27, pages 192–195. World Academy of Science.

M. Fiszman, D. Demner-Fushman, H. Kilicoglu, and T. C. Rindflesch. 2009. Automatic summarization of medline citations for evidence-based medical treatment: A topic-oriented evaluation. *Jouranl of Biomedical Informatics*, 42(5):801–813.

G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):1–39.

Y. Gong and X. Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM.

S. P. Hobson, B. J. Dorr, C. Monz, and R. Schwartz. 2007. Task-based evaluation of text summarization using relevance prediction. *Information Processing & Management*, 43(6):1482–1499.

R. Katragadda, P. Pingali, and V. Varma. 2009. Sentence position revisited: a robust light-weight update summarization 'baseline' algorithm. In *CLIAWS3 '09: Proceedings of the Third International Workshop on Cross Lingual Information Access*, pages 46–52, Morristown, NJ, USA. ACL.

C. Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26.

F. Liu and Y. Liu. 2008. Correlation between rouge and human evaluation of extractive meeting summaries. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 201–204. ACL.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165.

B. Maegaard, M. Atiyya, K. Choukri, S. Krauwer, C. Mokbel, and M. Yaseen. 2008. Medar: Collaboration between european and mediterranean arabic partners to support the development of language technology for arabic. In *In Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *Proceeding of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*. Association for Computational Linguistics.

R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. 2008. Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics.

J. Steinberger and K. Ježek. 2004. Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of the 5th International Conference on Information Systems Implementation and Modelling (ISIM)*, pages 93–100.

J.-Y. Yeh, H.-R. Ke, and W.-P. Yang. 2008. iSpreadRank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications*, 35(3):1451 – 1462.

DefArabicQA: Arabic Definition Question Answering System

Omar Trigui[1], Lamia Hadrich Belguith[1] , Paolo Rosso[2]
[1] ANLP Research Group- MIRACL Laboratory, University of Sfax, Tunisia
[2] Natural Language Engineering Lab. - ELiRF, Universidad Politécnica de Valencia, Spain
{omar.trigui,l.belguith}@fsegs.rnu.tn, prosso@dsic.upv.es

Abstract

Today the Web is the largest resource of knowledge and, therefore, sometimes this makes it difficult to find precise information. Current search engines can only return ranked snippets containing the effective answers to a query user. But, they can not return the exact answers. Question Answering systems present the solution to obtain effective and exact answers to a user question asked in natural language question instead of keywords query. Unfortunately, Question Answering task for the Arabic language has not been investigated enough in the last decade, compared to other languages. In this paper, we tackle the definition Question Answering task for the Arabic language. We propose an Arabic definitional Question Answering system based on a pattern approach to identify exact and accurate definitions about organization using Web resources. We experimented this system using 2000 snippets returned by Google search engine and Wikipedia Arabic version and a set of 50 organization definition questions. The obtained results are very encouraging: (90%) of the questions used have complete (vital) definitions in the top-five answers and (64%) of them have complete definitions in the top-one answer. MRR was (0.81).

1 Introduction

Definition questions of the type 'What is X?' is frequently asked on the Web. This type of question is generally asked for information about organization or thing. Generally, dictionaries and encyclopaedias are the best resources for this type of answers. However, these resources often do not contain the last information about a specific organization or do not yet contain a definition of a new organization due to non instantaneous update. Thus, the user has the habit to look for a definition from searching the Web. Our research takes place in this context to make easy the obtaining of the organization definition from Web resources. In this paper, we present a definitional Question Answering (QA) system for the Arabic language called *DefArabicQA*. This system outperforms the use of Web searching by two criteria: (i) permits to ask by an ordinary question (e.g., 'What is X?') instead of asking by keywords query; (ii) returns an accurate answer instead of mining the Web searching results in order to find the expected information.

The paper is organized as follows: Section 2 provides an overview of the Arabic QA systems. Section 3 presents our definitional QA system *DefArabicQA*. Section 4 presents the realized experiments and Section 5 discusses the obtained results. A conclusion and some future directions for our work are exposed in Section 6.

2 Related works

QA systems are designed to retrieve the exact answers from a set of knowledge resources to the user question. Many researches are interested in this task in many competitions (e.g., TREC[1], CLEF[2] and NTCIR[3]). An analysis of the TREC QA task experiments shows that two kinds of questions are mainly involved: factual and definition questions. A factual question is a simple fact retrieval where the answer is often a named entity (e.g. 'Who is the president of the League of Arab States?'). Whereas a definition question is a question asking for any important information about someone or something (e.g., 'What is the League of Arab States?'). Unfortunately, the evaluation platforms of QA task in the mainly evaluation conferences do not include the Arabic language. To our knowledge, no research has been done on Arabic definitional QA systems. However, there are some attempts to build factual QA systems (e.g. Hammo et al.,2002; Benajiba et al.,2007a; Brini et al.,2009). We cited below an overview of these factual Question Answering systems. (Hammo et al., 2002; 2004) developed *QARAB* a factual QA system. They employed information retrieval techniques to identify candidate passages, and sophisticated natural language processing techniques to parse the question and the top 10 ranked passages. They adopted a keyword matching strategy to identify answers. The answer identified is the whole sentence matching the question keywords. The evaluation process of this system was based on 113 questions and a set of documents collected from the newspaper Al-Raya. They obtained a precision equal to 97.3%, recall equal to 97.3% and MRR equal to 0.86 (Hammo et al.,2004). The average length of the answers obtained was 31 words. (Kanaan et al.,2004) developed a QA system using approximately the same method of (Hammo et

[1] Text Retrieval Conference http://trec.nist.gov/

[2] Cross-Language Evaluation Forum http://clef-campaign.org/

[3] NII Test Collection for IR Systems http://research.nii.ac.jp/ntcir/

al.,2002) system's. Their evaluation was based on a set of 25 documents from the Web and 12 questions. (Benajiba et al.,2007a) developed 'ArabiQA' a factual QA system. They employed *Arabic-JIRS*[4] (Benajiba et al.,2007b), a passage retrieval system to search the relevant passages. They used also the named entity system *ANERsys* (Benajiba et al.,2007c) to identify and classify named entities within the passages retrieved. The test-set consists of 200 questions and 11,000 documents from Wikipedia Arabic version. They reached a precision of 83.3% (Benajiba et al.,2007a). (Brini et al.,2009) developed a prototype to build an Arabic factual Question Answering system using Nooj platform[5] to identify answers from a set of education books. Most of these researches cited above, have not made test-bed publicly available, which makes it impossible to compare their evaluation results.

As we have already said, there is not a research focused on definitional QA systems for the Arabic language. Therefore, we have considered that an effort needs to be done in this direction. We built an Arabic QA system, which we named *DefArabicQA* that identifies and extracts the answers (i.e., exact definitions) from Web resources. Our approach is inspired from researches that have obtained good results in TREC experiments. Among these researches we cite the work of (Grunfeld & Kwok, 2006) which is based on techniques from IR, pattern matching and metakeyword detection with little linguistic analysis and no natural language understanding.

3 The *DefArabicQA* system

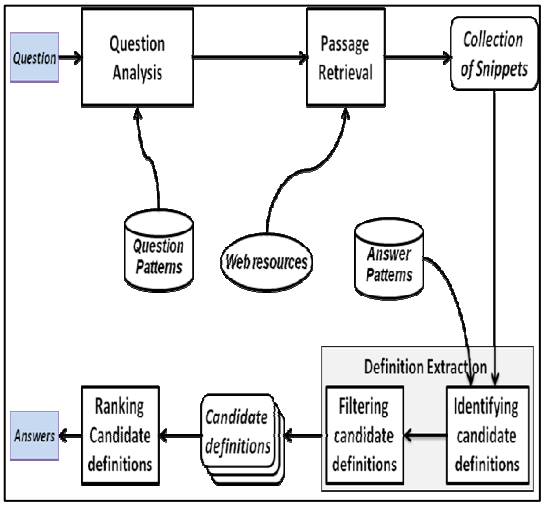

Figure. 1. Architecture of *DefArabicQA*

[4] http://sourceforge.net/projects/jirs/

[5] http://www.nooj4nlp.net/pages/nooj.html

The architecture of the *DefArabicQA* system is illustrated in Figure 1. From a general viewpoint, the system is composed of the following components: *i)* question analysis, *ii)* passage retrieval, *iii)* definition extraction and *iv)* ranking candidate definitions.

This system does not use any sophisticated syntactic or semantic techniques, as those used for factual QA systems (Hammo et al.,2002; Benajiba et al.,2007).

3. 1 Question analysis

This module is a vital component of *DefArabicQA*. The result of this module is the identification of the topic question (i.e., named entity) and the dedication of the answer type expected. The question topic is identified by using two lexical question patterns (Table. 1) and the answer type expected is deduced from the interrogative pronoun of the question.

Question patterns		Expected answer types
Who+be+<topic> ?	من هوا من هي <الموضوع>؟	Person
What+be+<topic> ?	ما هوا ما هي< الموضوع>؟	Organization

Table 1. Question patterns and their expected answer types used by *DefArabicQA* system

3.2 Passage retrieval

The passage retrieval module collects the top-n snippets retrieved by the Web search engine. This specific query is constituted of the question topic which is identified by the question analysis module. After collecting the top-n snippets, only those snippets containing the integrate question topic are kept on the basis of some heuristic (e.g. length of a snippet must be more than 13 characters).

3.3 Definition extraction

This module is the core module of a definitional QA system and it is composed of two sub-modules that are in charge of: *i)* identifying candidate definitions, and *ii)* filtering candidate definitions.

3.3.1. Identifying candidate definitions

In this step, we identify and extract candidate definitions from the collection of snippets collected in the passage retrieval module. We use lexical patterns to identify these candidate definitions. Generally, a lexical pattern is a sequence of strings (e.g., words, letters and punctuation symbols) which provide a context to identify the exact answers. It reflects a common use of written styles used to introduce an organization.

In our context, patterns are created manually and no natural language processing is employed in their construction. A candidate definition is identified by a specific pattern if the surrounding of the question

topic in a snippet is recognized by a specific pattern.

3.3.2. Filtering candidate definitions

We use heuristic rules to filter the identified candidate definitions. These heuristic rules are deduced from the observation of a set of annotated candidate definitions (i.e., a collection of candidate definitions divided in incorrect candidate definitions and correct candidate definitions).

3.4 Definition ranking

The component "definition ranking" is based on a statistical approach. We used a global score to rank candidate definitions retained in the "Definition Extraction" module. This global score is a combination of three scores related to three criteria of a candidate definition: *i)* pattern weight criterion, *ii)* snippet position criterion, and *iii)* word frequency criterion. We present to the user the first top-5 candidate definitions ranked according to their global scores.

3.4.1. Pattern weight criterion (C_1)

The score of this criterion is the weight of the pattern that has identified the candidate definition CD_i. This score is represented by:

$$C_1(CD_i) = w_i \qquad (1)$$

Where w_i presents the weight of pattern *i*. We associate a weight to each pattern according to its relevance.

3.4.2. Snippets position criterion (C_2)

The score of this criterion represents the position of the snippet that contains the candidate definition (in the snippets collection). This score is done by:

$$C_2(CD_i) = p_i \qquad (2)$$

Where p_i is the snippet position containing the candidate definition CD_i.

3.4.3. Word frequency criterion (C_3)

The score of this criterion represents the sum of the frequencies of the words occurring in a candidate definition. According to this criterion, the candidate definition CD_i score is calculated as follows. Firstly, we construct a centroid vector containing common words across candidate definitions with their frequencies, beyond stopwords. Secondly, we calculate the frequency sum of the words recurring in both CD_i and centroid vector as indicated by the following formulate:

$$C_3(CD_i) = \sum_{k=1}^{n} f_{ik} \qquad (3)$$

Where n is the number of words which occur in the centroid vector and in the candidate definition CD_i, $1 \le k \le n$ and f_{ik} is the frequency of word$_k$.

3.4.4. Criteria aggregation

In order to aggregate the three criteria described above, we first proceed to the normalization of the score of each criterion by dividing it by the maximum score as follows:

$$C'_{i,j} = C_{i,j} / MaxC_i \qquad (4)$$

Where *i* is a candidate definition and *j* a criterion. Then, we combine the three normalized scores in order to obtain the global score *GS* of the candidate definition CD_i. This global score is obtained by:

$$GS(CD_i) = \sum_{j=1}^{3} C'_{i,j} \qquad (5)$$

4. Experiments and results

This section describes two experiments carried out using the *DefArabicQA* system. The first experiment was carried out using Google Search engine[6], while the second experiment was carried out using Google Search engine and the free encyclopedia Wikipedia Arabic version[7]. In both experiments, we used 50 organization definition questions[8] similar to these used in TREC. The system was assessed by an Arabic native speaker. As evaluation metrics, we use MRR. It is a measure used in TREC QA section and it is calculated as follows: each question is assigned a score equal to the inverse rank of the first string that is judged to contain a correct answer. If none of the five answer strings contain an answer, the question is assigned a score of zero. The MRR value for the experiment is calculated by taking the average of scores for all the questions (Voorhees, 2001).

4.1 Results of the first experiment

Out of the 50 questions in the test collection 41 questions (82%) were answered correctly by complete definitions in the top-five candidate definitions. 54% of the questions were answered by the first candidate definition returned, 14% by the second candidate definition, 6% by the third candidate definition, 6% by the fourth candidate

[6] http://www.google.com/intl/ar/

[7] http://ar.wikipedia.org/w/index.php?title=خاص:بحث&search=&go=اذهب

[8] Resources available for research purpose at: http://sites.google.com/site/omartrigui/downloads

definition, 2% by the fifth candidate definition as shown in Table 2. The systems missed 18% of the questions as shown in Table 3. MRR was equal to 0.70 as shown in Table 4.

4.2 Results of the second experiment

The main goal of the second experiment is to measure the value added by the Web resource Wikipedia to the results obtained in the first experiment with the Google search engine.

In this experiment, we used the same set of questions of the first experiment with Google search engine and Wikipedia as Web resources. Out of the 50 questions in the test collection, 45 questions (90%) were answered correctly by complete definitions in the top-five candidate definitions. 64% of the questions were answered by the first returned candidate definition, 16% by the second candidate definition, 4% by the third candidate definition, 2% by the fourth candidate definition and 4% by the fifth candidate definition as shown in Table 2. The system missed 10% of the questions as shown in Table 3. The obtained value of MRR is 0.81 (see Table 4).

	Experiment I	**Experiment II**
Rank 1st	27 **(54%)**	32 **(64%)**
Rank 2nd	7 (14%)	8 (16%)
Rank 3th	3 (6%)	2 (4%)
Rank 4th	3 (6%)	1 (2%)
Rank 5th	1 (2%)	2 (4%)
Top-five	41 **(82%)**	45 **(90%)**

Table 2. Rate of the answered questions for each Rank (the Top-5 positions)

	Experiment I	**Experiment II**
Top-5	9 (18%)	5 (10%)

Table 3. Rate of non answered questions (in the Top-5 positions)

	Experiment I	**Experiment II**
MRR	**0.70**	**0.81**

Table 4. MRR values for both experiments

5. Discussion

The two experiments cited above showed that our approach applied in *DefArabicQA* system returned reasonably good results.

The Web resource Wikipedia has improved the results of *DefArabicQA* when it was coupled with Google in the second experiment. The MRR was increased from 0.70 (in the first experiment) to 0.81 (in the second experiment) and the rate of non answered question in the Top-5 positions was decreased from 18% (in the first experiment) to 10% (in the second experiment). Also, the Rate of the questions answered by the first returned candidate definition was increased from 54% (in the first experiment) to 64% (in the second experiment).

6. Conclusion and future work

In this paper we proposed a definitional Question Answering system called *DefArabicQA*. This system provides effective and exact answers to definition questions expressed in Arabic language from Web resources. It is based on an approach which employs a little linguistic analysis and no language understanding capability. *DefArabicQA* identifies candidate definitions by using a set of lexical patterns, filters these candidate definitions by using heuristic rules and ranks them by using a statistical approach.

Two evaluation experiments have been carried out on *DefArabicQA*. The first experiment was based on Google as a Web resource and has obtained an MRR equal to 0.70 and a rate of questions answered by the first answer equal to 54%, while the second experiment was based on Google coupled with Wikipedia as Web resources. In this experiment, we obtained an MRR equal to 0.81 and a rate of questions answered by the first answer equal to 64%. 50 definition questions are used for both experiments.

As future works, we plan to improve the quality of the definitions when it is truncated. Indeed, in some cases, few words are missed at the end of the definition answer. This is due to the fact that the snippet itself is truncated. As a solution, we will download the original Web page and segment the useful snippet correctly using a tokenizer. We also plan to conduct an empirical study to determine different weights to the three used criteria for ranking the candidate definitions. These weights will reflect the importance of each criterion.

Acknowledgments

This research work started thanks to the bilateral Spain-Tunisia research project on "Answer Extraction for Definition Questions in Arabic" (AECID-PCI B/017961/08).

The work of the third author was carried out in the framework of the AECID-PCI C/026728/09 and the TIN2009-13391-C04-03 research projects.

References

Benajiba, Y., Rosso, P., and Lyhyaoui, A. (2007.a). Implementation of the ArabiQA Question Answering System's Components. In *Proceedings of Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium*.

Benajiba, Y., Rosso, P., and J.M. Gomez. (2007.b). Adapting the JIRS Passage Retrieval System to the Arabic Language. In *Proceeding of CICLing conference, Springer-Verlag, 2007.* pages 530--541.

Benajiba, Y., Rosso., P. and Benedi Ruiz. J.M. (2007.c). ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. *In Proceeding of CICLing conference, volume 4394 of Lecture Notes in Computer Science,* Springer-Verlag, 2007. pages 143--153.

Brini, W., Ellouze, M., Mesfar, S., Hadrich Belguith, L. (2009). An Arabic Question-Answering system for factoid questions. In *Proceeding of IEEE International Conference on Natural Language Processing and Knowledge Engineering*. pages 797--805.

Hammou, B., Abu-salem, H., Lytinen, S., and Evens. M. (2002). QARAB: A question answering system to support the Arabic language. In *Proceedings of the workshop on computational approaches to Semitic languages*, *ACL*, pages 55--65.

Hammo, B., Ableil, S., Lytinen S., and Evens, M. (2004). Experimenting with a question answering system for the Arabic language. In *Computers and the humanities,* vol. 38, N°4, pages 397--415.

Kanaan, G., Hammouri, A., Al-Shalabi R., and Swalha. M. (2009). A New Question Answering System for the Arabic Language. In *American Journal of Applied Sciences* 6 (4), pages 797--805.

Grunfeld, L., and Kwok, K.L. (2006). Sentence ranking using keywords and meta-keywords. In: Advances in Open Domain Question Answering, T. Strzalkowski and S. Harabagiu, (eds.). Springer, pages 229--258.

Voorhees, E., (2001). Overview of the TREC 2001 Question Answering Track. In *Proceedings of the 10th Text REtrieval Conference*, pages 42–51.

Techniques for Arabic Morphological Detokenization and Orthographic Denormalization

Ahmed El Kholy and Nizar Habash

Center for Computational Learning Systems, Columbia University
475 Riverside Drive New York, NY 10115
{akholy,habash}@ccls.columbia.edu

Abstract

The common wisdom in the field of Natural Language Processing (NLP) is that orthographic normalization and morphological tokenization help in many NLP applications for morphologically rich languages like Arabic. However, when Arabic is the target output, it should be properly detokenized and orthographically correct. We examine a set of six detokenization techniques over various tokenization schemes. We also compare two techniques for orthographic denormalization. We discuss the effect of detokenization and denormalization on statistical machine translation as a case study. We report on results which surpass previously published efforts.

1. Introduction

Arabic is a morphologically rich language. The common wisdom in the field of natural language processing (NLP) is that tokenization of Arabic words through decliticization and reductive orthographic normalization is helpful for many applications such as language modeling and statistical machine translation (SMT). Tokenization and normalization reduce sparsity and decrease the number of out-of-vocabulary (OOV) words. However, in order to produce proper Arabic that is orthographically correct, tokenized and orthographically normalized words should be detokenized and orthographically corrected (enriched). As an example, the output of English-to-Arabic machine translation (MT) systems is reasonably expected to be proper Arabic regardless of the preprocessing used to optimize the MT performance. Anything less is comparable to producing all lower-cased English or uncliticized and undiacritized French. Detokenization is not a simple task because there are several morphological adjustments that apply in the process. In this paper we examine different detokenization techniques for various tokenization schemes and their effect on SMT output as a case study.

This paper is divided as follows. Section 2 presents the previous related work. In Section 3, we discuss the Arabic linguistic issues and complexities that motivate the detokenization techniques explained in Section 4. Section 5 describes the various experiments we had followed by an analysis of the results.

2. Related Work

Much work has been done on Arabic-to-English MT (Habash and Sadat, 2006; Lee, 2004; Zollmann et al., 2006) mostly focusing on reducing the sparsity caused by Arabic's rich morphology. There is also a growing number of publications with Arabic as target language. In previous work on Arabic language modeling, OOV reduction was accomplished using morpheme-based models (Heintz, 2008). Diehl et al. (2009) also used morphological decomposition for Arabic language modeling for speech recognition. They described an SMT approach to detokenization (or what they call morpheme-to-word conversion). Although the implementation details are different, their solution is comparable to one of our new (but not top performing) decomposition models (T+LM). We do not compare directly to their implementation approach in this paper. Regarding English-to-Arabic MT, Sarikaya and Deng (2007) use joint morphological-lexical language models to re-rank the output English-dialectal Arabic MT; and Badr et al. (2008) report results on the value of morphological tokenization of Arabic during training and describe different techniques for detokenization of Arabic in the output. The research presented here is most closely related to that of Badr et al. (2008). We extend on their contribution and present a comparison of a larger number of tokenization schemes and detokenization techniques that yield improved results over theirs.

3. Arabic Linguistic Issues

In this section, we present relevant aspects of Arabic word orthography and morphology.

3.1. Arabic Orthography

Certain letters in Arabic script are often spelled inconsistently which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (same form corresponding to multiple words). In particular, variants of Hamzated Alif, أ Â[1] or إ Ă are often written without their Hamza (ء '): ا *A*; and the Alif-Maqsura (or dotless Ya) ى ý and the regular dotted Ya ي *y* are often used interchangeably in word final position. This inconsistent variation in raw Arabic text is typically addressed in Arabic NLP through what is called orthographic normalization, a reductive process that converts all Hamzated Alif forms to bare Alif and dotless Ya/Alif Maqsura form to dotted Ya. We will refer to this kind of normalization as a Reduced normalization (RED). We introduce a different type of normalization that selects the appropriate form of the Alif. We call this Enriched normalization (ENR). ENR Arabic is optimally the desired correct form of Arabic to generate.

[1] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007).

Comparing a manually enriched (ENR) version of the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) to its reduced (RED) version, we find that 16.2% of the words are different. However, the raw version of the PATB is only different in 7.4% of the words. This suggests a major problem in the recall of the correct ENR form in raw text.

Another orthographic issue is the optionality of diacritics in Arabic script. In particular, the absence of the Shadda diacritic (ـّ ∼) which indicates a doubling of the consonant it follows leads to a different number of letters in the tokenized and untokenized word forms (when the tokenization happens to split the two doubled consonants). See the example in Table 1 under (Y-Shadda). Consequently, the detokenization task for such cases is not a simple string concatenation.

3.2. Arabic Morphology

Arabic is a morphologically complex language with a large set of morphological features producing a large number of rich word forms. While the number of (morphologically untokenized) Arabic words in a parallel corpus is 20% less than the number of corresponding English words, the number of unique Arabic word types is over twice the number of unique English word types over the same corpus size.

One aspect of Arabic that contributes to this complexity is its various attachable clitics. We define three degrees of cliticization that are applicable in a strict order to a word base:

```
[cnj+ [prt+ [art+ BASE +pro]]]
```

At the deepest level, the BASE can have either the definite article (+ال *Al+* 'the') or a member of the class of pronominal enclitics, +pro, (e.g., +هم *+hm* 'their/them'). Next comes the class of particle proclitics (prt+), e.g., +ل *l+* 'to/for'. At the shallowest level of attachment we find the conjunction proclitic (cnj+), e.g., +و *w+* 'and'. The attachment of clitics to word forms is not a simple concatenation process. There are several orthographic and morphological adjustment rules that are applied to the word. An almost complete list of these rules relevant to this paper are presented and exemplified in Table 1.

It is important to make the distinction here between simple word segmentation, which splits off word substrings with no orthographic/morphological adjustments, and tokenization, which does. Although segmentation by itself can have important advantages, it leads to the creation of inconsistent or ambiguous word forms: consider the words مكتبة *mktbħ* 'library' and مكتبهم *mktbthm* 'their library'. A simple segmentation of the second word creates the non-word string مكتبت *mktbt*; however, applying adjustment rules as part of the tokenization generates the same form of the basic word in the two cases. For more details, see (Habash, 2007). In this paper, we do not explore morphological tokenization beyond decliticization.

4. Approach

We would like to study the value of a variety of detokenization techniques over different tokenization schemes and orthographic normalization. We report results on naturally occurring Arabic text and English-Arabic SMT outputs. To that end, we consider the following variants:

4.1. Tokenization

We consider five tokenization schemes discussed in the literature, in addition to a baseline no-tokenization scheme (D0). The D1, D2, TB and D3 schemes were first presented by Habash and Sadat (2006) and the S2 scheme was presented by Badr et al. (2008). The S1 scheme used by Badr et al. (2008) is the same as Habash and Sadat (2006)'s D3 scheme. TB is the PATB tokenization scheme. We use the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow, 2005) to produce the various tokenization schemes. The schemes are presented in Table 2 with various relevant statistics. The schemes differ widely in terms of the increase of number of tokens and the corresponding type count reduction. The more verbose schemes, i.e., schemes with more splitting, have lower out-of-vocabulary (OOV) rates and lower perplexity but are also harder to predict correctly.

4.2. Detokenization

We compare the following techniques for detokenization:

- Simple (S): concatenate clitics to word without applying any orthographic or morphological adjustments.
- Rule-based (R): use deterministic rules to handle all of the cases described in Table 1. We pick the most frequent decision for ambiguous cases.
- Table-based (T): use a lookup table mapping tokenized forms to detokenized forms. The table is based on pairs of tokenized and detokenized words from our language model data which had been processed by MADA. We pick the most frequent decision for ambiguous cases. Words not in the table are handled with the (S) technique. This technique essentially selects the detokenized form with the highest conditional probability $P(detokenized|tokenized)$.
- Table+Rule(T+R): same as (T) except that we back off to (R) not (S).

The above four techniques are the same as those used by Badr et al. (2008). We introduce two new techniques that use a 5-gram untokenized-form language model and the `disambig` utility in the SRILM toolkit (Stolcke, 2002) to decide among different alternatives:

- T+LM: we use all the forms in the (T) approach. Alternatives are given different conditional probabilities, $P(detokenized|tokenized)$, derived from the tables. Backoff is the (S) technique. This technique essentially selects the detokenized form with the highest $P(detokenized|tokenized) \times P_{LM}(detokenized)$.
- T+R+LM: same as (T+LM) but with (R) as backoff.

Rule Name	Condition	Result	Example		
Definite Article	?ل+ال+ل *l+Al+l?*	+لل *ll+*	ل+ال+مكتب *l+Al+mktb*	للمكتب *llmktb*	‘for the office’
			ل+ال+لجنة *l+Al+ljnħ*	للجنة *lljnħ*	‘for the committee’
Ta-Marbuta	ة- *-ħ* +pron	ت- *-t* +pron	مكتبة+هم *mktbħ+hm*	مكتبتهم *mktbthm*	‘their library’
Alif-Maqsura	ى- *-ý* +pron	ا- *-A* +pron	روى+ه *rwY+h*	رواه *rwAh*	‘he watered it’
	exceptionally	ي- *-y* +pron	على+ه ς*lY+h*	عليه ς*lyh*	‘on him’
Waw-of-Plurality	وا- *-wA* +pron	و- *-w* +pron	كتبوا+ه *ktbwA+h*	كتبوه *ktbwh*	‘they wrote it’
	تم- *-tm* +pron	تمو- *-tmw* +pron	كتبتم+ه *ktbtmw+h*	كتبتموه *ktbtmwh*	‘you [pl.] wrote it’
Hamza	ء- *-’* +pron	ئ- *-ŷ* +pron	بهاء+ه *bhA’+h*	بهائه *bhAŷh*	‘his glory [gen.]’
	less frequently	ؤ- *-ŵ* +pron	بهاء+ه *bhA’+h*	بهاؤه *bhAŵh*	‘his glory [nom.]’
	less frequently	ء- *-’* +pron	بهاء+ه *bhA’+h*	بهاءه *bhA’h*	‘his glory [acc.]’
Y-Shadda	ي+ي- *-y* +*y*	ي *y*	قاضي+ي *qADy+y*	قاضيّ *qADy*	‘my judge’
N-Assimilation	من *mn* +*m/n*	م *m* +*m/n*	من+ما *mn+mA*	ممـا *mmA*	‘from which’
	عن ς*n* +*m/n*	ع ς +*m/n*	عن+من ς*n+mn*	عمن ς*mn*	‘about whom’
	لا+أن *Ân* +*lA*	ألّا *ÂlA*	أن+لا *Ân+lA*	ألّا *ÂlA*	‘that ... not’

Table 1: Orthographic and Morphological Adjustment Rules

	Definition	Change Relative to D0			Prediction Error Rate			OOV		Perplexity	
		Token#	ENR Type#	RED Type#	ENR	RED	SEG	ENR	RED	ENR	RED
D0	word				0.62	0.09	0.00	2.22	2.17	412.3	410.6
D1	cnj+ word	+7.2	-17.6	-17.8	0.76	0.23	0.14	1.91	1.89	259.3	258.2
D2	cnj+ prt+ word	+13.3	-32.3	-32.6	0.89	0.37	0.25	1.50	1.50	185.5	184.7
TB	cnj+ prt+ word +pro	+17.9	-43.9	-44.2	1.07	0.57	0.42	1.22	1.22	142.2	141.5
S2	cnj+prt+art word +pro	+40.6	-53.0	-53.3	1.20	0.73	0.60	0.91	0.91	69.3	69.0
D3	cnj+ prt+ art+ word +pro	+44.2	-53.0	-53.3	1.20	0.73	0.60	0.90	0.90	61.9	61.7

Table 2: A comparison of the different tokenization schemes studied in this paper in terms of their definition, the relative change from no-tokenization (D0) in tokens (Token#) and enriched and reduced word types (ENR Type# and RED Type#), MADA’s error rate in producing the enriched tokens, the reduced tokens and just segmentation (SEG); the out-of-vocabulary (OOV) rate; and finally the perplexity value associated with different tokenization. OOV rates and perplexity values are measured against the NIST MT04 test set while prediction error rates are measured against a Penn Arabic Treebank devset.

4.3. Normalization

We consider two kinds of orthographic normalization schemes, enriched Arabic (ENR) and reduced Arabic (RED). For tokenized enriched forms, the detokenization produces the desired output. In case of reduced Arabic, we consider two alternatives to automatic orthographic enrichment. First, we use MADA to enrich Arabic text after detokenization (MADA-ENR). MADA can predict the correct enriched form of Arabic words at 99.4%.[2] Alternatively, we jointly detokenize and enrich using detokenization tables that map reduced tokenized words to their enriched detokenized form (Joint-DETOK-ENR).

In terms of evaluation, we report our results in both reduced and enriched Arabic forms. We only compare in the matching form, i.e., reduced hypothesis to reduced reference and enriched hypothesis to enriched reference.

[2]Statistics are measured on a devset from the Penn Arabic Treebank (Maamouri et al., 2004).

5. Experimental Results

5.1. Detokenization

We compare the performance of the different detokenization techniques discussed in Section 4. for the ENR and the RED normalization conditions. The performance of the different techniques is measured against the Arabic side of the NIST MT evaluation set for 2004 and 2005 (henceforth, MT04+MT05) which together have 2,409 sentences comprising 64,554 words. We report the results in Table 3 in terms of sentence-level detokenization error rate defined as the percentage of sentences with at least one detokenization error. The best performer across all conditions is the T+R+LM technique. The previously reported best performer was T+R (Badr et al., 2008), which was only compared with D3 and S2 tokenizations only.

As illustrated in the results, the more complex the tokenization scheme, the more prone it is to detokenization errors. Moreover, RED has equal or worse results than ENR under all conditions except for the S detokenization technique with the TB, S2 and D3 schemes. This is a result of the S

detokenization technique not performing any adjustments, which leads to the never-word-internal Alif-Maqsura character appearing incorrectly in word-internal positions in ENR. While for RED, the Alif-Maqsura is reductively normalized to Ya, which is the correct form in some of the cases.

The results for S2 and D3 are identical because these two schemes only superficially differ in whether proclitics are space-separated or not. Similarly, TB results are identical to D3 for the S and R techniques. This can be explained by the fact that the only difference between the D3 and TB schemes is that the definite article is attached to the word (in TB and not D3), a difference that does not produce different results under the deterministic S and R techniques.

We analyze the errors (14 cases) for the T+R+LM technique on D3 scheme and classify them into two categories. The first category comprises 11 cases ($\approx$ 80% of the errors) and is caused by ambiguity resulting from the lack of diacritical marks. Seven (50% overall) of these errors involve the selection of the correct Hamza form before a pronominal enclitic. For example, the tokenized word وأشقاء+ها *w+Âšqa'+hA* 'and+siblings+her' can be detokenized to وأشقاءها *wÂšqa'hA* or وأشقائها *wÂšqAŷhA* or وأشقاؤها *wÂšqAŵhA* depending on the grammatical case of the noun أشقاء *Âšqa'*, which is only expressible as a diacritical mark. The other four cases involve two closed class words, إن *Ăn* and لكن *lkn*, each of which corresponding to two diacritized forms that require different adjustments. For example, the tokenized word إن+ني *Ăn+ny* can be detokenized to إني *Ăny* (إن+ني *Ăin+niy* $\rightarrow$ إنِّي *Ăin∼iy*) or إنني *Ănny* (إنَّ+ني *Ăin∼a+niy* $\rightarrow$ إنَّني *Ăin∼aniy*). In many cases, the n-gram language model is able to select for the correct form, but it is not always successful. The second category of errors compromises 3 cases ($\approx$ 20% of the errors) which involve automatic tokenization failures producing tokens that are impossible to map back to the correct detokenized form.

5.2. Orthographic Enrichment and Detokenization

As previously mentioned, it's desirable for Arabic-generating automatic applications to produce orthographically correct Arabic. As such, reduced tokenized output should be enriched and detokenized to produce proper Arabic. We compare next the two different enrichment techniques discussed in Section 4.: using MADA to enrich detokenized reduced text (MADA-ENR) versus detokenizing and enriching in one joint step (Joint-DETOK-ENR). We consider the effect of applying these two techniques together with the various detokenization techniques when possible. The comparison is presented for D3 in Table 4. D3 has the highest number of tokens per word and it's the hardest to detokenize as shown in Table 3. The MADA-ENR enrichment technique can be applied to the output of all detokenization techniques; however, the Joint-DETOK-ENR enrichment technique can only be used as part of table-based detokenization techniques. The results for basic ENR and RED detokenization are in columns two and three. Columns four and five present the two approaches to enriching the tokenized reduced text. Although the Joint-DETOK-ENR technique does not outperform MADA-ENR for T and T+R, it significantly benefits from the use of the LM extension to these two techniques. In fact, Joint-DETOK-ENR produces the best results overall under T+R+LM, with an error rate that is 20% lower than the best performance by MADA-ENR. Overall, however, enriching and detokenizing RED text yields output that has almost 10 times the error rate compared to detokenizing ENR. This is expected since ENR is far less ambiguous than RED. The best performer across all conditions for detokenization and enrichment is the T+R+LM approach.

All experiments reported so far in this paper start with a perfect pairing between the original and tokenized words. The real challenge is applying the detokenization techniques on automatically produced (noisy) text. The next section discusses the effect of detokenization on SMT output as a case study.

5.3. Tokenization and Detokenization for SMT

In this section we present English-to-Arabic SMT as a case study for the effect of tokenization in improving the quality of translation. Then, we show the performance of the different detokenization techniques on the output and their reflections over the overall performance of the SMT systems.

5.3.1. Experimental Data

All of the training data we use is available from the Linguistic Data Consortium (LDC).[3] We use an English-Arabic parallel corpus of about 142K sentences and 4.4 million words for translation model training data. The parallel text includes Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18). Lemma based word alignment is done using GIZA++ (Och and Ney, 2003). For language modeling, we use 200M words from the Arabic Gigaword Corpus (LDC2007T40) together with the Arabic side of our training data. Twelve language models were built for all combinations of normalization and tokenization schemes. We used 5-grams for all LMs unlike (Badr et al., 2008) who used different n-grams sizes for tokenized and untokenized variants. All LMs are implemented using the SRILM toolkit (Stolcke, 2002).

MADA is used to preprocess the Arabic text for translation modeling and language modeling. MADA produced all enriched forms and tokenizations. Due to the fact that the number of tokens per sentence changes from one tokenization scheme to another, we filter the training data so that all experiments are done on the same number of sentences. We use the D3 tokenization scheme as a reference and set the cutoff at 100 D3 tokens. English preprocessing simply included down-casing, separating punctuation from words and splitting off "'s".

All experiments are conducted using the Moses phrase-based SMT system (Koehn et al., 2007). The decoding weight optimization was done using a set of 300 sentences from the 2004 NIST MT evaluation test set (MT04). The

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

	S		R		T		T+R		T+LM		T+R+LM	
	ENR	RED	ENR	RED	ENR	RED	ENR	RED	ENR	RED	ENR	RED
D1	0.17	0.17	0.17	0.17	0.08	0.08	0.08	0.08	0.08	0.08	**0.08**	**0.08**
D2	22.50	22.50	0.58	0.79	0.37	0.37	0.21	0.21	0.37	0.37	**0.21**	**0.21**
TB	38.36	35.53	1.41	3.03	1.33	1.49	0.75	0.91	1.16	1.25	**0.58**	**0.66**
S2	38.36	35.53	1.41	3.03	1.37	1.54	0.79	0.95	1.20	1.29	**0.62**	**0.71**
D3	38.36	35.53	1.41	3.03	1.37	1.54	0.79	0.95	1.20	1.29	**0.62**	**0.71**

Table 3: Detokenization results in terms of sentence-level detokenization error rate.

Detokenization	ENR	RED		
	ENR	RED	MADA-ENR	Joint-DETOK-ENR
S	38.36	35.53	39.73	
R	1.41	3.03	10.59	
T	1.37	1.54	8.92	9.46
T+R	0.79	0.95	8.68	9.22
T+LM	1.20	1.29	9.34	6.23
T+R+LM	**0.62**	**0.71**	**7.39**	**5.89**

Table 4: Detokenization and enrichment results for D3 tokenization scheme in terms of sentence-level detokenization error rate.

tuning is based on the tokenized Arabic without detokenization. We use a maximum phrase length of size 8 for all experiments. We report results on the 2005 NIST MT evaluation set (MT05). These test sets were created for Arabic-English MT and have 4 English references. We use only one Arabic reference in reverse direction for both tuning and testing. We evaluate using BLEU-4 (Papineni et al., 2002) although we are aware of its caveats (Callison-Burch et al., 2006).

5.3.2. Tokenization Experiments

System	ENR		RED	
Evaluation	ENR	RED	ENR	RED
D0	24.63	24.67	24.66	24.71
D1	25.92	25.99	26.06	26.12
D2	26.41	26.49	26.06	26.15
TB	**26.46**	**26.51**	**26.73**	**26.80**
S2	25.71	25.76	26.11	26.19
D3	25.68	25.75	25.03	25.10

Table 5: Comparing different tokenization schemes for statistical MT in BLEU scores over detokenized Arabic (using T+R+LM technique)

We compare the performance of the different tokenization schemes and normalization conditions. The results are presented in Table 5 using T+R+LM detokenization technique. The best performer across all conditions is the TB scheme. The previously reported best performer was S2 (Badr et al., 2008), which was only compared against D0 and D3 tokenizations. Our results are consistent with Badr et al. (2008)'s results regarding D0 and D3. However, our TB result outperforms S2. The differences between TB and all other conditions are statistically significant above the 95% level. Statistical significance is computed using paired bootstrap resampling (Koehn, 2004). Training over RED Arabic then enriching its output sometimes yields better results than training on ENR directly which is the case with the TB tokenization scheme. However, sometimes the opposite is true as demonstrated in the D3 results. This is due to the tradeoff between the quality of translation and the quality of detokenization which is discussed in the next section.

5.3.3. Detokenization Experiments

We measure the performance of the different detokenization techniques discussed in Section 4. against the SMT output for the TB tokenization scheme. We report results in terms of BLEU scores in Table 6. The results for basic ENR and RED detokenization are in columns two and three. Column four presents the results for the Joint-DETOK-ENR approach to joint enriching and detokenization of tokenized reduced output discussed in Section 4.

When comparing Table 6 (in BLEU scores) with the corresponding cells in Table 4 (in sentence-level detokenization error rate), we observe that the wide range of performance in Table 4 is not reflected in BLEU scores in Table 6. This is expected given the different natures of the tasks and metrics used. Although the various detokenization techniques do not preserve their relative order completely, the S technique remains the worst performer and T+R+LM remains the best in both tables. However, the R and T+LM techniques perform relatively much better with MT output than they do with naturally occurring text. The most interesting observation is perhaps that under the best performing T+R+LM technique, joint detokenization and enrichment (Joint-DETOK-ENR) outperforms ENR detokenization despite the fact that Joint-DETOK-ENR has over nine times the error rate in Table 4. This shows that improved MT quality using RED training data out-weighs the lower quality of automatic enrichment.

Detokenization	ENR	RED	
	ENR	RED	Joint-DETOK-ENR
S	25.57	26.04	N/A
R	26.45	26.78	N/A
T	26.40	26.78	22.44
T+R	26.40	26.78	22.44
T+LM	26.46	26.80	26.73
T+R+LM	**26.46**	26.80	**26.73**

Table 6: BLEU scores for SMT outputs with different detokenization techniques over TB tokenization scheme

5.3.4. SMT Detokenization Error Analysis

Since we do not have a gold detokenization reference for our MT output, we automatically identify detokenization errors resulting in non-words (i.e., invalid words). We analyze the SMT output for the D3 tokenization scheme and T+R+LM detokenization technique using the morphological analyzer component in the MADA toolkit,[4] which provides all possible morphological analyses for a given word and identifies words with no analysis. We find 94 cases of words with no analysis out of 27,151 words (0.34%), appearing in 84 sentences out of 1,056 (7.9%). Most of the errors come from producing incompatible sequences of clitics, such as having a definite article with a pronominal clitic. For instance, the tokenized word ال+علاقة+نا *Al+ςlAqħ+nA* 'the+relation+our' is detokenized to العلاقتنا *AlςlAqtnA* which is grammatically incorrect. This is not a detokenization problem per se but rather an MT error. Such errors could still be addressed with specific detokenization extensions such as removing either the definite article or the pronominal clitic.

6. Conclusions and Future Work

We presented experiments studying six detokenization techniques to produce orthographically correct and enriched Arabic text. We presented results on naturally occurring Arabic text and MT output against different tokenization schemes. The best technique under all conditions is T+R+LM for both naturally occurring Arabic text and MT output. Regarding enrichment, joint enrichment with detokenization gives better results than performing the two tasks in two separate steps. Moreover, the best setup for MT is training on RED text and then enriching and detokenizing the output using the joint technique.

In the future, we plan to investigate the creation of mappers trained on seen examples in our tables to produce ranked detokenized alternatives for unseen tokenized word forms. In addition, we plan to examine language modeling approaches that target Arabic's complex morphology such as factored LMs (Bilmes and Kirchhoff, 2003). We also plan to explore ways to make detokenization robust to MT errors.

[4]This component uses the databases of the Buckwalter Arabic Morphological Analyzer (Buckwalter, 2004).

7. Acknowledgement

The work presented here was funded by a Google research award. We would like to thank Ioannis Tsochantaridis, Marine Carpuat, Alon Lavie, Hassan Al-Haj and Ibrahim Badr for helpful discussions.

8. References

Ibrahim Badr, Rabih Zbib, and James Glass. 2008. Segmentation for english-to-arabic statistical machine translation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 153–156, Columbus, Ohio, June. Association for Computational Linguistics.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference/North American Chapter of Association for Computational Linguistics (HLT/NAACL-03)*, pages 4–6, Edmonton, Canada.

T. Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 249–256, Trento, Italy.

F. Diehl, M.J.F. Gales, M. Tomalin, and P.C. Woodland. 2009. Morphological Analysis and Decomposition for Arabic Speech-to-Text Systems. In *Proceedings of InterSpeech*.

Nizar Habash and Owen Rambow. 2005. 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, June. Association for Computational Linguistics.

Nizar Habash and Fatiha Sadat. 2006. Arabic Preprocessing Schemes for Statistical Machine Translation. In *Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL06)*, pages 49–52, New York, NY.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Nizar Habash. 2007. Arabic Morphological Representations for Machine Translation. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Ilana Heintz. 2008. Arabic language modeling with finite state transducers. In *Proceedings of the ACL-08: HLT Student Research Workshop*, pages 37–42, Columbus, Ohio, June. Association for Computational Linguistics.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2004. Statistical significance tests formachine translation evaluation. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP'04)*, Barcelona, Spain.

Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, pages 57–60, Boston, MA.

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

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–52.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Ruhi Sarikaya and Yonggang Deng. 2007. Joint morphological-lexical language modeling for machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 145–148, Rochester, New York, April. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO.

Andreas Zollmann, Ashish Venugopal, and Stephan Vogel. 2006. Bridging the inflection morphology gap for arabic statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 201–204, New York City, USA. Association for Computational Linguistics.

Tagging Amazigh with AnCoraPipe

Mohamed Outahajala, Lahbib Zekouar, Paolo Rosso, M. Antònia Martí

Institut Royal de la Culture Amazighe, Ecole Mohammadia des Ingénieurs, Natural Language Engineering Lab – EliRF(DSIC), CLiC - Centre de Llenguatge i Computació

Avenue Allal El Fassi, Madinat Al Irfane - Rabat - Instituts Adresse postale : BP 2055 Hay Riad Rabat Morocco, Avenue Ibnsina B.P. 765 Agdal Rabat Morocco, Universidad Politécnica de Valencia, Spain, Universitat de Barcelona 08007 Barcelona, Spain

E-mail: outahajala@ircam.ma, zenkouar@emi.ac.ma, prosso@dsic.upv.es, amarti@ub.edu

Abstract

Over the last few years, Moroccan society has known a lot of debate about the Amazigh language and culture. The creation of a new governmental institution, namely IRCAM, has made it possible for the Amazigh language and culture to reclaim their rightful place in many domains. Taking into consideration the situation of the Amazigh language which needs more tools and scientific work to achieve its automatic processing, the aim of this paper is to present the Amazigh language features for a morphology annotation purpose. Put in another way, the paper is meant to address the issue of Amazigh's tagging with the multilevel annotation tool AnCora Pipe. This tool is adapted to use a specific tagset to annotate Amazigh corpora with a new defined writing system. This step may well be viewed as the first step for an automatic processing of the Amazigh language; the main aim at very beginning being to achieve a part of speech tagger.

Introduction

Amazigh (Berber) is spoken in Morocco, Algeria, Tunisia, Libya, and Siwa (an Egyptian Oasis); it is also spoken by many other communities in parts of Niger and Mali. It is a composite of dialects of which none have been considered the national standard used by tens of millions of people in North Africa mainly for oral communication.

With the emergence of an increasing sense of identity, Amazigh speakers would very much like to see their language and culture rich and developed. To achieve such a goal, some Maghreb states have created specialized institutions, such as the Royal Institute for Amazigh Culture (IRCAM, henceforth) in Morocco and the High Commission for Amazigh (HCA) in Algeria. In Morocco, Amazigh has been introduced in mass media and in the educational system in collaboration with relevant ministries. Accordingly, a new Amazigh television channel was launched in first mars 2010 and it has become common practice to find Amazigh taught in various Moroccan schools as a subject.

Over the last 7 years, IRCAM has published more than 140 books related to the Amazigh language and culture, a number which exceeds the whole amount of Amazigh publications in the 20th century, showing the importance of an institution such as IRCAM. However, in Natural Language Processing (NLP) terms, Amazigh, like most non-European languages, still suffers from the scarcity of language processing tools and resources.

In this sense, since morphosyntactic tagging is an important and basic step in the processing of any given language, the main objective of this paper is to explain how we propose to supply the Amazigh language with this important tool.

For clarity reasons, this paper is organized as follows: in the first part we present an overview of the Amazigh language features. Then, we provide a brief retrospective on Amazigh morphology as conceived by IRCAM linguists. Next we give an overview on Amazigh corpora. The fourth section describes how to tag with AnCoraPipe and the fifth section deals with Amazigh tagset.

2. The Amazigh language

Amazigh belongs to the Hamito-Semitic/"Afro-Asiatic" languages (Cohen 2007, Chaker 1989) with rich templatic morphology. In linguistic terms, the language is characterized by the proliferation of dialects due to historical, geographical and sociolinguistic factors. In Morocco, one may distinguish three major dialects: Tarifit in the North, Tamazight in the center and Tashlhiyt in the southern parts of the country; 50% of the Moroccan population speak Amazigh (Boukouss, 1995), but according to the last governmental demolinguisitc data of 2004, the Amazigh language was spoken only by some 28% of the Moroccan population (around 10 Million inhabitants), showing an important decrease of its use.

Amazigh standardization cannot be achieved without adopting a realistic strategy that takes into consideration its linguistic diversity (Ameur et al., 2006a; Ameur et al. 2006b). As far as the alphabet is concerned, and because of historical and cultural reasons, Tifinaghe has become the official graphic system for writing Amazigh. IRCAM kept only pertinent phonemes for Tamazight, so the number of the alphabetical phonetic entities is 33, but Unicode codes only 31 letters plus a modifier letter to form the two phonetic units: ⵗⵯ(g^w) and ⴽⵯ(k^w). The whole range of Tifinagh letters is subdivided into four subsets: the letters used by IRCAM, an extended set used also by IRCAM, other neo-tifinaghe letters in use and some attested modern Touareg letters. The number reaches 55 characters (Zenkouar 2004, Andries 2004). In order to rank strings and to create keyboard layouts for Amazigh in accordance with international standards, two other standards have been adapted (Outahajala and Zenkouar, 2004):

- ISO/IEC14651 standard related to international string

ordering and comparison method for comparing character strings and description of the common template tailorable ordering;
- Part 1: general principles governing keyboard layouts of the standard ISO/IEC 9995 related to keyboard layouts for text and office systems.
Most Amazigh words may be conceived of as having consonantal roots. They can have one, two, three or four consonants, and may sometimes extend to five. Words are made out of these roots by following a pattern (Chafiq 1991). For example the word 'aslmad' is built up from the root *lmd* "study" by following the pattern as12a3, where the number 1 is replaced by the first consonant of the root, number 2 is replaced by the second consonant of the root and number 3 is replaced by the 3rd consonant of the root.
Concerning spelling, the system put by IRCAM is based on a set of rules and principles applied to "words" along which the parsing of pronounced speech into written separated words is effected. A grapheme, a written word, according to the spelling system is a succession of letters which can sometimes be one letter delimited by whitespace or punctuation.
The graphic rules for Amazigh words are set out as follows (Ameur et al 2006a, 2006b, Boukhris et al 2008):
- Nouns consist of a single word occurring between two blank spaces. To the noun are attached the morphological affixes of gender (masculine/ feminine), number (singular/plural) and state (free/construct) as it is shown in the following examples: ⵄⵎⴰⵢ/ⵜⵄⵎⴰⵢⵜ (amzday (masc.)/**t**amzday**t**(fem.)) "a dweller", ⵄⵎⴰⵢ/ⵉⵎⴰⵢⵏ (amzday (sing.)/**i**mzday**n**(plr.)) "dweller/dwellers", and ⵄⵎⴰⵢ/ⵓⵎⴰⵢ (amzday (free state)/umzday (construct state)). Kinship names constitute a special class since they are necessarily determined by possessive markers which form with them one word, for example: ⴱⴰⴱⴾ (babak) which means "your father";
- Quality names/adjectives constitute a single word along with the morphological indicators of gender (masculine/ feminine), number (singular/plural), and state (free/construct);
- Verbs are single graphic words along with its inflectional (person, number, aspect) or derivational morphemes. For example: ⵜⵜⴰⵣⵣⵍ /ttazzl/which means "you run (imperfective)". The verb is separated by a blank space from its predecessor and successor pronouns, i.e.: ⵢⴰⵙⵉ ⵜⵍ / ⴰⵏ ⵜⵍ ⵢⴰⵙⵉ ("yasi tn / ad tn yasi" which means "he took them / he will take them");
- Pronouns are isolated from the words they refer to. Pronouns in Amazigh are demonstrative, exclamative, indefinite, interrogative, personal, possessive, or relative. For instance, ⴰⵏ (ad) in the phrase ⴰⴱⵓⵉⵏ ⴰⵏ (abrid ad), which means "this way", is an example of a demonstrative pronoun;
- An adverb consists of one word which occurs between two blank spaces. Adverbs are divided into adverbs of place, time, quantity, manner, and interrogative adverbs.
- Focus mechanisms, interjections and conjunctions are written in the form of single words occurring between two blank spaces. An example of a conjunction is: ⵎⵓ (mr) which means "if";
- Prepositions are always an independent set of characters with respect to the noun they precede; however, if the preposition is followed by a pronoun, both the preposition and the noun make a single whitespace-delimited string. For example: ⵖⵓ (γr) "to, at" + ⵉ (i) "me" possessive pronoun gives ⵖⴰⵓⵉ/ⵖⵯⵓⵉ (γari/γuri) "to me, at me, with me";
- Particles are always isolated. There are aspect particles such as ⴰⵣⵣⴰ (aqqa), ⴰⵓ (ar), ⴰⵏ (ad), particles of negation such as ⵯⵓ (ur), orientation particles like ⵍⵍ in ⴰⵍⵉ ⵍⵍ!(awi nn) "take it there" and a predicative particle ⵏ (d);
- Determinants take always the form of single between two blank spaces. Determiners are divided into articles, demonstratives, exclamatives, indefinite articles, interrogatives, numerals, ordinals, possessives, presentatives, quantifiers. ⴽⵯⵍⵍⵯ (kullu) "all" is a quantifier for instance;
- Amazigh punctuation marks are similar to the punctuation marks adopted in international languages and have the same functions. Capital letters, nonetheless, do not occur neither at the beginning of sentences nor at the initial of proper names.
The English terminology used above was extracted form (Boumalk and Naït-Zerrad, 2009).

3. Amazigh corpora

3.1 Amazigh corpora features

Amazigh corpora have the following characteristics:
- They are extracted from geographically circumscribed dialects;
- Some varieties are less represented than others, or not studied at all;
- There is special need for a more general type of work whose goal is to collect the data of all dialects;
- Existing publications are scattered and inaccessible in most cases. Some of them go back to the XIXth century and the beginning of the XXth century. The few existing copies of those references are only available in specialized libraries, mainly in France;
- General documents containing the data of all Amazigh dialects do not exist (phonetics, semantics, morphology, phraseology…etc.).
- Some existing texts need revision because of segmentation problems.
To constitute an annotated corpus, we have chosen a list of corpora extracted from the Amazigh version of IRCAM's web site[1], the periodical *Inghmisn n usinag*[2] (IRCAM newsletter) and school textbooks. We were able to reach a total number of words superior to 20k words. A comparative quantity of corpora was used in tagging other languages, for example (Allauzen and Bonneau-Maynard, 2008).

[1] www.ircam.ma
[2] Freely downloadable from http://www.ircam.ma/amz/index.php?soc=bulle

3.2 Writing systems

Amazigh corpora produced up to now are written on the basis of different writing systems, most of them use Tifinaghe-IRCAM (Tifinaghe-IRCAM makes use of Tifinaghe glyphs but Latin characters) and Tifinaghe Unicode. It is important to say that the texts written in Tifinaghe Unicode are increasingly used.

Even though, we have decided to use a specific writing system based on ASCII characters for the following reasons:

- To have a common set of characters for annotated corpora;
- To facilitate texts treatment for annotators since ASCII characters are known by all systems;
- To handle its use due to the fact that people are still more familiar with Arabic and Latin writing systems.

In Table 1 of correspondences between the different writing systems and transliteration correspondences is shown

Tifinaghe Unicode		Transliteration		Used characters in Tifinaghe IRCAM		Chosen characters for tagging
Code	Character	Latin	Arabic	characters	codes	
U+2D30	ⴰ	a	ا	A, a	65, 97	a
U+2D31	ⴱ	b	ب	B, b	66, 98	b
U+2D33	ⴳ	g	گ	G, g	71, 103	g
U+2D33& U+2D6F	ⴳ ⵯ	g^w	گ + ◌ٗ	Å, å	197, 229	g°
U+2D37	ⴷ	d	د	D, d	68, 100	d
U+2D39	ⴹ	ḍ	ض	Ä, ä	196, 228	D
U+2D3B	ⴻ	e^3	يـــ	E, e	69, 101	e
U+2D3C	ⴼ	f	ف	F, f	70, 102	f
U+2D3D	ⴽ	k	ک	K, k	75, 107	k
U+2D3D& U+2D6F	ⴽ ⵯ	k^w	گ + ◌ٗ	Æ, æ	198, 230	k
U+2D40	ⵀ	h	ه	H, h	72,104	h
U+2D40	ⵃ	ḥ	ح	P, p	80,112	H
U+2D44	ⵄ	ɛ	ع	O, o	79, 111	E
U+2D45	ⵅ	x	خ	X, x	88, 120	x
U+2D47	ⵇ	q	ق	Q, q	81, 113	q
U+2D49	ⵉ	i	ي	I, i	73, 105	i
U+2D4A	ⵊ	j	ج	J, j	74, 106	j
U+2D4D	ⵍ	l	ل	L, l	76, 108	l
U+2D4E	ⵎ	m	م	M, m	77, 109	m
U+2D4F	ⵏ	n	ن	N, n	78, 110	n
U+2D53	ⵓ	u	و	W, w	87, 119	u
U+2D54	ⵔ	r	ر	R, r	82, 114	r
U+2D55	ⵕ	ṛ	ڕ	Ë, ë	203, 235	R
U+2D56	ⵖ	ɣ	غ	V, v	86, 118	G
U+2D59	ⵙ	s	س	S, s	83, 115	s
U+2D5A	ⵚ	ṣ	ص	Ã, ã	195, 227	S
U+2D5B	ⵛ	c	ش	C, c	67, 99	c
U+2D5C	ⵜ	t	ت	T, t	84, 116	t
U+2D5F	ⵟ	ṭ	ط	Ï, ï	207, 239	T
U+2D61	ⵡ	w	ۏ	W, w	87, 119	w
U+2D62	ⵢ	◌ٔ + ي	ي	Y, y	89, 121	y
U+2D63	ⵣ	z	ز	Z, z	90, 122	z
U+2D65	ⵥ	ẓ	ژ	Ç, ç	199, 231	Z
U+2D6F	ⵯ	ʷ	◌ٗ	No correspondant in Tifinaghe-IRCAM		°

Table1: The mapping from existing writing systems and the chosen writing system.

A transliteration tool was build in order to handle transliteration to and from the chosen writing system and to correct some elements such as the character "^" which exists in some texts due to input error in entring some Tifinaghe letters. So the sentence portion "ⵄⵙⵙ ⵉ ⵜⵍⵢⵓⵄ" using Tifinaghe Unicode or "ass n tmvra" using Tifinaghe-IRCAM will be transliterated as "ass n tmGra" ("When the day of the wedding arrives").

4. AnCoraPipe tool

AnCoraPipe (Bertran et al. 2008) is a corpus annotation tool which allows different linguistic levels to be annotated efficiently, since it uses the same format for all stages. The tool reduces the annotation time and makes easy the integration of the different annotators and the different annotation levels.

The input documents may have a standard XML format, allowing to represent tree structures (specially usefull at syntactic anotation stages). As XML is a wide spread standard, there are many tools available for its analysis, transformation and management.

AnCoraPipe includes an integrated search engine based on XPath language (http://www.w3.org/TR/xpath/), which allows to find structures of all kinds among the documents. For corpus analysis, an export tool can summarize the attributes of all nodes in the corpus in a grid that can easily be imported to basic analysis tools (such as Excel or OpenOffice calc), statistical software (SPSS) or Machine Learning tools (Weka).

A default tagset is provided in the standard installation. It has been designed as generic as possible in order to match the requisites of a wide amount of languages. In spite of that, if the generic tagset is not useful, the interface is fully customizable to allow different tagsets defined by the user.

In order to allow AnCoraPipe usable in a full variety of languages, the user can change the visualization font. This may help viewing non-latin scripts such as Chinese, Arabic or Amazigh.

AnCoraPipe is currently an Eclipse Plugin. Eclipse is an extendable integrated development environment. With this plugin, all features included in Eclipse are made available

[3] note : different use in the IPA which uses the letter ə

for corpus annotation and developing. In particular, the Eclipse's collaboration and team plugins can be used to organize the work of a group of annotators.

5. AnCoraPipe for Amazigh

AnCoraPipe allows the definition of different tagsets. We have decided to work with a set of ASCII characters for the following reasons:
- Amazigh text corpora are written in different writing systems;
- Amazigh linguists are still familiar with Latin alphabets;
- the default tagset is a multilevel tagset;
- to simplify the interface for linguists;
- to avoid adding some tags which are not currently needed as co-reference tags, syntactic tags...etc.

Based on the Amazigh language features presented above, Amazigh tagset may be viewed to contain 13 nodes with two common attributes to each node: "wd" for "word" and "lem" for "lemma", whose values depend on the lexical item they accompany.

Amazigh nodes and their attributes are set out in what follows:

PoS	**attributes and subattributes with number of values**
Noun	gender(3), number(3), state(2), derivative(2),PoSsubclassification(4), person(3), possessornum(2), possessorgen(2)
Adjective/ name of quality	gender(3), number(3), state(2), derivative(2), PoS subclassification(3)
Verb	gender(3), number(3), form(5), aspect(3), negative(2), form(2)
Pronoun	gender(3), number(3), PoS subclassification(7), deictic(3), autonome(2), person(3), possessornum(2), possessorgen(2)
Determiner	gender(3), number(3), PoS subclassification(11)
Adverb	PoS subclassification(5)
Preposition	gender(3), number(3), PoS subclassification(6), person(3), possessornum(2), possessorgen(2)
Conjunction	PoS subclassification(2)
Interjection	
Particle	PoS subclassification(5)
Focus	
Residual	PoS subclassification(5), gender(3), number(3)
Punctuation	punctuation mark type(16)

Table2: A synopsis of the features of the Amazigh PoS tagset with their attributes and values

In Table 2 the node Residual stands for attributes like currency, number, date, math marks and other unknown residual words.

Manual annotation is being carried out by a team of linguists. Technically, manual annotation proceeds along the requirements of the tool presented above.

A sample of annotated Corpora as presented in Section 3:

Here follows the annotation of a sentence extracted from a text about a wedding ceremony:

"ass n tmGra, iwsn asn ayt tqbilt. illa ma issnwan, illa ma yakkan i inbgiwn ad ssirdn"

[English translation: "When the day of the wedding arrives, the people of the tribe help them. Some of them cook; some other help the guests get their hands washed "]

```
<sentence>
<n gen="m" lem="ass" num="s" state="free" wd="ass"/>
<prep wd="n"/>
<n gen="f" lem="tamGra" num="s" state="construct" wd="tmGra"/>
<pu punct="comma" wd=","/>
<v aspect="perfective" gen="m" lem="aws" num="p" person="3" wd="iwsn"/>
<p gen="m" num="p" person="3" postype="personal" wd="asn"/>
<d gen="m" num="p" postype="indefinite" wd="ayt"/>
<n gen="f" lem="taqbilt" num="s" postype="common" state="construct" wd="tqbilt"/>
<pu punct="period" wd="."/>
<v aspect="perfective" gen="m" lem="ili" num="s" person="3" wd="illa"/>
<p postype="relative" wd="ma"/>
<v aspect="imperfective" gen="m" lem="ssnw" num="s" person="3" form="participle" wd="issnwan"/>
<pu punct="comma" wd=","/>
<v aspect=" perfective" gen="m" lem="ili" num="s" person="3" wd="illa"/>
<p postype="relative" wd="ma"/>
<v aspect="imperfective" form="participle" gen="m" lem="fk" num="s" person="3" wd="yakkan"/>
<prep wd="i"/>
<n gen="m" lem="anbgi" num="p" state="construct" wd="inbgiwn"/>
<pr postype="aspect" wd="ad"/>
<v aspect="aorist" gen="m" lem="ssird" num="p" person="3" wd="ssirdn"/>
```

The main aim of this corpus is to achieve a part of speech tagger based on Support Vector Machines (SVM) and Conditional Random Fields (CRF) because they have been proved to give good results for sequence classification (Kudo and Matsumoto, 2000, Lafferty et al. 2001). We are planning to use freely available tools like Yamcha and

CRF++ toolkits[4].

6. Conclusion and future works

In this paper, after a brief description about social and linguistic characteristics of the Amazigh language, we have addressed the basic principles we followed for tagging Amazigh written corpora with AnCoraPipe: the tagset used, the transliteration and the annotation tool.

In the future, it is our goal to tag more corpora to constitute a reference corpus for works on Amazigh NLP and we plan also to work on Amazigh Base Phrase Chunking.

Acknowledgments

We would like to thank Manuel Bertran for improving the AnCora Pipe tool to support Amazigh features, all IRCAM researchers and Professor Iazzi El Mehdi from Ibn Zohr University, Agadir for their explanations and precious help. The work of the last two authors was carried out thanks to AECID-PCI C/026728/09 and TIN2009-13391-C04-03/04 research projects.

References

Allauzen, A. Bonneau-Maynard, H. (2008).Training and evaluation of POS taggers on the French MULTITAG corpus. In proceedings of LREC 08.

Ameur, M., Boujajar, A., Boukhris, F. Boukouss, A., Boumaled, A., Elmedlaoui, M., Iazzi, E., Souifi, H. (2006a), *Initiation à la langue Amazighe*. Publications de l'IRCAM. pp. 45—77.

Ameur, M., Boujajar, A., Boukhris, F. Boukouss, A., Boumaled, A., Elmedlaoui, M., Iazzi, E. (2006b) *Graphie et orthographe de l'Amazighe*. Publications de l'IRCAM.

Andries, P. (2004). La police open type Hapax berbère. In proceedings of the workshop : *la typographie entre les domaines de l'art et l'informatique*, pp. 183—196.

Bertran, M., Borrega, O., Recasens, M., Soriano, B. (2008). AnCoraPipe: A tool for multilevel annotation. *Procesamiento del lenguaje Natural*, nº 41. Madrid (Spain).

Boukhris, F. Boumalk, A. El moujahid, E., Souifi, H. (2008). *La nouvelle grammaire de l'Amazighe*. Publications de l'IRCAM.

Boukhris, F. (2006). Structure morphologique de la préposition en Amazighe. In proceedings of the workshop: *Structures morphologiques de l'Amazighe*. Publications de l'IRCAM. pp. 46-56.

Boukouss, A. (1995). Société, langues et cultures au Maroc: Enjeux symboliques, *publications de la Faculté des Lettres de Rabat*.

Boumalk, A., Naït-Zerrad, K. (2009). *Amawal n tjrrumt -Vocabulaire grammatical*. Publications de l'IRCAM.

Chafiq, M. (1991) .أربعة وأربعون درسا في الأمازيغية. éd. Arabo-africaines.

Chaker, S. (1989). Textes en linguistique berbère - introduction au domaine berbère, éditions du CNRS, 1984. P 232-242.

Cohen, D. (2007). Chamito-sémitiques (langues). In *Encyclopædia Universalis*.

Iazzi, E., Outahajala,M. (2008), Amazigh Data Base. In proceedings of LREC 08.

Kudo, T., Yuji Matsumoto, Y. (2000). Use of Support Vector Learning for Chunk Identification.

Lafferty, J. McCallum, A. Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In proceedings of ICML-01 282-289

Outahajala, M., Zenkouar, L. (2005). La norme du tri, du clavier et Unicode. In proceedings of the workshop : *la typographie entre les domaines de l'art et l'informatique*, pp. 223—238.

Saa, F. (2006). Les thèmes verbaux de l'Amazighe. In proceedings of the workshop: *Structures morphologiques de l'Amazighe*, pp.102--111.

Zenkouar, L. (2004). L'écriture Amazighe Tifinaghe et Unicode, in *Etudes et documents berbères*. Paris (France). n° 22, pp. 175—192.

Zenkouar, L. (2008). Normes des technologies de l'information pour l'ancrage de l'écriture Amazighe, in *Etudes et documents berbères*. Paris (France), n° 27, pp. 159—172.

[4] Freely downloadable from http://chasen.org/~taku/software/YamCha/ and http://crfpp.sourceforge.net/

Verb Morphology of Hebrew and Maltese — Towards an Open Source Type Theoretical Resource Grammar in GF

Dana Dannélls* **and John J. Camilleri†**

*Department of Swedish Language, University of Gothenburg
SE-405 30 Gothenburg, Sweden
†Department of Intelligent Computer Systems, University of Malta
Msida MSD2080, Malta
dana.dannells@svenska.gu.se, jcam0003@um.edu.mt

Abstract

One of the first issues that a programmer must tackle when writing a complete computer program that processes natural language is how to design the morphological component. A typical morphological component should cover three main aspects in a given language: (1) the lexicon, i.e. how morphemes are encoded, (2) orthographic changes, and (3) morphotactic variations. This is in particular challenging when dealing with Semitic languages because of their non-concatenative morphology called root and pattern morphology. In this paper we describe the design of two morphological components for Hebrew and Maltese verbs in the context of the Grammatical Framework (GF). The components are implemented as a part of larger grammars and are currently under development. We found that although Hebrew and Maltese share some common characteristics in their morphology, it seems difficult to generalize morphosyntactic rules across Semitic verbs when the focus is towards computational linguistics motivated lexicons. We describe and compare the verb morphology of Hebrew and Maltese and motivate our implementation efforts towards a complete open source type theoretical resource grammars for Semitic languages. Future work will focus on semantic aspects of morphological processing.

1. Introduction

One of the first issues that a programmer must tackle when writing a complete computer program that processes natural language is how to design the morphological component. A typical morphological component should cover three main aspects in a given language: (1) the lexicon, i.e. how morphemes are encoded, (2) orthographic changes, and (3) morphotactic variations. This is in particular challenging when dealing with Semitic languages because of their non-concatenative morphology called root and pattern morphology (Goldberg, 1994).

The Grammatical Framework (GF) is a grammar formalism for multilingual grammars and their applications (Ranta, 2004). It has a Resource Grammar Library (Ranta, 2009) that is a set of parallel natural language grammars that can be used as a resource for various language processing tasks. Currently, the only Semitic morphological component included in the library is for Arabic (Dada and Ranta, 2007). To increase the coverage of Semitic languages we decided to develop two additional resource grammars for Hebrew and Maltese. The availability of several languages belonging to the same language family in one framework fosters the development of common language modules where grammatical rules across languages are generalised. Thus, increasing the potential of yielding interesting insights highlighting similarities and differences across languages. These kind of modules already exist in GF for Romance and Scandinavian languages.

In this paper we describe our implementations of Hebrew and Maltese verb morphologies in the context of GF. We present how two of the three morphological aspects mentioned above are accounted departing from the similarities and differences of verb formation in each of the two languages.

2. Verb morphology

Each of the Semitic languages has a set of verbal patterns, which is a sequence of vowels (and possibly consonants) into which root consonants are inserted. The root itself has no definite pronunciation until combined with a vocalic pattern, i.e. a template. The combination of morphological units is non-linear, i.e. it relies on intertwining between two independent morphemes (root and pattern).[1]

There are different ways in how templates modify the root consonants: doubling the middle consonants, inserting vowels between consonants, adding consonantal affixes, etc. Inflectional morphology systems are constructed by attaching prefixes and suffixes to lexemes. Verb lexemes are inflected for person, number, gender and tense. Common tenses of Semitic languages are: present, perfect, imperfect, and imperative.[2]

2.1. Modern Hebrew

Hebrew has seven verb pattern groups (binyanim) that are associated with a fixed morphological form, e.g. pa'al: C1aC2aC3, nif'al: niC1C2aC3, pi'el:C1iC2eC3. There are two major root classifications: regular (strong) and irregular (weak). In the same manner that each verb belongs to a particular binyan, it also belongs to a particular group of verbs (Hebrew gzarot) that classify them by their root composition (for an extensive information about the Hebrew root and pattern system see Arad (2005)). For regular verbs, all root consonants are present in all the verb forms, there are fixed rules that distinguish how verbs are

[1] Linguists consider the root to be a morpheme despite the fact that it is not a continuous element in the word, and it is not pronounceable (McCarthy, 1979; McCarthy, 1981).

[2] In Semitic languages, the past tense is referred by the term perfect and the future tense by imperfect.

conjugated depending on their guttural root letters, i.e. A, h, H, O.[3] An example of two verbs that are conjugated differently in pa'al future tense because in one of the verbs the root's second guttural letter is *h* are: Agmwr (g.m.r, 'will finish') with *w* stem vowel, and Anhag (n.h.g, 'will drive') with *a* stem vowel. Irregular verbs are verbs where one or more of the root consonants are either missing or altered, which causes some deviation from a fully regular conjugation. These verbs can be classified into three main groups, each of which contains three to five subgroups (Coffin and Bolozky, 2005). Roots are conjugated differently depending on the root classification group they belong to, e.g. ywred (y.r.d, pa'al, group2_py, 'he goes down'), yasen (y.s.n, pa'al, group3_py, 'he sleeps'). The sub-groups of irregular verbs contain large root composition variations which depend on the different occurrences of the root's consonants; phonological changes contribute to these irregularities in verbs forms and inflections.

The Hebrew binyanim are associated with a semantic trait. This leads to a certain complexity when designing a morphological component. What leads to this complexity is the fact that some morphemes (roots) are combined with more than one pattern, resulting in ambiguity problem. On the other hand not all roots are realised in all patterns. To avoid inefficient parsing that generates too many results, that in turn introduces new difficulties in identifying the root's consonants and in resolving ambiguities, it is necessary to employ semantic markings in the lexicon.

2.2. Maltese

Maltese verb pattern groups (themes) are a subset Classical Arabic pattern groups. These patterns involve affixation and prefixation, for example, niżżel (theme II, 'bring down'), tniżżel (theme V, 'be brought down'). Verbs in theme I must be specified as undergoing a vowel change which is always $a \rightarrow o$ or $e \rightarrow o$. Theme II is defined by double middle radical, the vowel possibilities are fixed. Most of the Semitic Maltese verbal themes exhibit the same properties that can be seen in theme I and II.

The vowels of the Semitic Maltese verb templates, unlike those of Classical Arabic, do not have a fixed vowel pattern, rather a vast range of vowel patterns. Each template allows several different vowel patterns determined by the tense and person of conjugation. For example, the root's template *ħ-d-m*, under perfect 3rd person singular, takes the pattern *a-a* (ħadem), whilst the past participle takes the pattern *i-a* (ħidma).

Each verb stem has two vowels and there are seven different verb types. Since a very large number of Maltese verbs are borrowed from Romance (Sicilian and Italian) and English, the productive verbal morphology is mainly affixal with a concatenative nature (Hoberman and Aronoff, 2003). The synchronic, productive processes of verb derivation, has resulted in three distinctive verb morphology features that are often referred in terms of: Semitic Maltese, Romance Maltese, and English Maltese (Mifsud, 1995).

Roots can be classified into one of five groups: strong, weak, defective, hollow, double and quadriliteral (4 radicals instead of 3). A root bears a semantic meaning that is converted into passive, active, reflexive forms depending on the pattern it belongs, e.g. *ħ-r-ġ* 'out', ħriġna 'we went out'.

Conjugations have predictable patterns and it is possible to predict the patterns and the entire conjugation tables from a given verb form (Aquilina, 1960; Aquilina, 1962). This may motivate the choice of representing lexemes in the lexicon (Ussishkin and Twist, 2007).

3. The Grammatical Framework (GF)

The Grammatical Framework is a functional grammar formalism based on Martin-Löf's type-theory (Martin-Löf, 1975) implemented in Haskell.

GF has three main module types: abstract, concrete, and resource. Abstract and concrete modules are top-level in the sense that they appear in grammars that are used at runtime for parsing and generation. One abstract grammar can have several corresponding concrete grammars; a concrete grammar specifies how the abstract grammar rules should be linearized in a compositional manner. A resource grammar is intended to define common parts of the concrete syntax in application grammars. It contains linguistic operations and parameters that are used to produce different forms and can be used as inherent features.

GF has a Resource Grammar Library, i.e. a set of parallel grammars that are built upon one abstract syntax. The GF's library, containing grammar rules for seventeen languages,[4] plays the role of a standard software library (Ranta, 2009). It is designed to gather and encapsulate morphological and syntactic rules of languages, which normally require expert knowledge, and make them available for non-expert application programmers by defining a complete set of morphological paradigms and a syntax for each language.

4. The grammar design

In this section we present how the verb morphologies of Modern Hebrew and Maltese are implemented in GF. The presented code fragments do not cover all aspect of the verb, such as passive/active mood, Hebrew infinitive form, Hebrew verbs with obligatory prepositions, English Maltese, etc. However, the code provides a glimpse of the two computational resources that are being developed. The presented code contains parameters, operations and lexicon linearizations which are defined according to GF's concrete and resource syntaxes. Parameters are defined to deal with agreement, operations are functions that form inflection tables, linearizations are string realisations of functions that are defined in the abstract syntax.

4.1. Common parameters

Both languages share the same parameter types and attributes for verbs, including: number (Singular, Plural),

[3] Throughout the paper we regulate the encoding of Hebrew characters using ISO-8859-8.

[4] The Resource Grammar Library currently (2010) contains the 17 languages: Arabic (complete morphology), Bulgarian, Catalan, Danish, Dutch, English, Finnish, French, German, Italian, Norwegian (bokmål), Polish, Romanian, Russian, Spanish, Swedish and Urdu.

gender (Masculine, Feminine), case (Nominative, Accusative, Genitive), person (first, second, third), voice (Active, Passive) and tense (Perfect, Participle, Imperfect). These types have the following definitions in GF syntax:

```
Number  = Sg | Pl ;
Gender  = Masc | Fem ;
Case    = Nom | Acc | Gen ;
Person  = P1 | P2 | P3 ;
Voice   = Active | Passive ;
Tense   = Perf | Part | Imperf ;
```

4.2. Modern Hebrew

An additional parameter *VPersonNumGen* provides a detailed description about how verbs are inflected. The parameter's attributes indicate: first person singular/plural, second and third person singular/plural and gender.

```
VPerNumGen = Vp1Sg | Vp1Pl | Vp2Sg Gender
  | Vp2Pl Gender | Vp3Sg Gender
  | Vp3Pl Gender ;
```

Operations

The Hebrew operations include: *Pattern*, i.e. a string consisting of a four position pattern slot, *Root*, i.e. a string consisting of either three or four (*Root4*) consonants. The Hebrew *Verb* is defined as a string that is inflected for tense person, number and gender. The *mkVPaal* operation defines regular verb paradigms for each tense and agreement features. The operation *getRoot* associates every consonant in the input string *v* with a variable. This is accomplished by the operation *C@?* which binds each consonant in the string *s* to a variable, e.g. C1 and C2. These variables are than coded into patterns using the operation *appPattern* which specifies how the root's consonants should be inserted into a pattern, given a root and a pattern.

```
Pattern : Type={C1, C1C2, C2C3, C3 : Str};
Root    : Type={C1,C2,C3 : Str};
Root4   : Type=Root ** {C4 : Str};
Verb    : Type={s : Tense ⇒ VPerNumGen ⇒ Str };

mkVPaal : Str → Verb = \v →
let root = getRoot v
in {s = table {
  Perf ⇒ table {
    Vp1Sg ⇒ appPattern root C1aC2aC3ty ;
    Vp1Pl ⇒ appPattern root C1aC2aC3nw ;
    Vp2Sg Masc ⇒ appPattern root C1aC2aC3th ;
    Vp2Sg Fem  ⇒ appPattern root
        C1aC2aC3t ;
    Vp2Pl Masc ⇒ appPattern root C1aC2aC3tM ;
    ... }
  Imperf ⇒ table { ... }
  }
};

getRoot : Str → Root = \s → case s of {
  C1@? + C2@? + C3 =>
    {C1 = C1 ; C2 = C2 ; C3 = C3}
};

appPattern : Root → Pattern → Str = \r,p →
    p.C1 + r.C1 + p.C1C2 + r.C2 + p.C2C3 + r.C3 +
         p.C3 ;
```

Patterns

Root patterns are defined in a separate resource. Patterns specify consonant slots and morphological forms, some examples are:

```
C1aC2aC3ty = {C1=""; C1C2=""; C2C3=""; C3="ty"};
C1aC2aC3nw = {C1=""; C1C2=""; C2C3=""; C3="nw"};
C1aC2aC3th = {C1=""; C1C2=""; C2C3=""; C3="th"};
```

Lexicon

Lexicon entries are functions that are defined in the abstract syntax. Below is an example of how the three verb entries: *write_V2*, *pray_V* and *sleep_V*, are linearized in the Hebrew lexicon. The lexicon generates verb paradigms through their binyanim, using the Hebrew operations.

```
write_V2 = mkVPaal "ktb" ;
pray_V   = mkVHitpael "pll" ;
sleep_V  = mkVPaalGroup3_py "ysn";
```

4.3. Maltese

There are additional parameters defined for Maltese, these include: VerbType (Strong, Defective, Weak, Hollow, Double), VOrigin (Semitic, Romance), VForm (for possible tenses, persons and numbers).

```
VType   = Strong | Defective | Weak | Hollow |
    Double ;
VOrigin = Semitic | Romance ;
VForm   = VPerf PerGenNum | VImpf PerGenNum |
    VImp Number ;
```

Operations

The operations for Maltese include: *Pattern*, i.e. a string consisting of two vowels, *Root*, i.e. a string consisting of four consonants of which one can be eliminated. The Maltese *Verb* is defined as a string inflected for tense, person, gender and number, that has the parameter values: *VerbType* and *VerbOrigin*. The *mkVerb* operation utilizes additional operations such as *classifyVerb, mkDefective, mkStrong* etc. to identify the correct verb. The operation *classifyVerb* takes a verb string and returns its root, pattern, and verb type, i.e. *Strong, Defective, Quad* etc. The operation *v1@#Vowel* matches the pattern *Vowel* and binds the variable *v1* to it. It is based on pattern matching of vowels.

```
Pattern : Type = {v1, v2 : Str} ;
Root : Type = {K, T, B, L : Str} ;
Verb : Type = {s : VForm ⇒ Str ; t : VType ; o :
     VOrigin} ;

mkVerb : Str → Verb = \mamma →
  let
    class = classifyVerb mamma
  in
    case class.t of {
      Strong ⇒ mkStrong class.r class.p ;
      Defective ⇒ mkDefective class.r class.p ;
      Quad ⇒ mkQuad class.r class.p ;
      ...
    } ;

classifyVerb : Str → { t:VType ; r:Root ;
  p:Pattern } = \mamma → case mamma of {
      K@#Consonant + v1@#Vowel
      + T@#Consonant + B@#Consonant
      + v2@#Vowel + L@#Consonant ⇒
```

```
      { t=Quad ; r={ K=K ; T=T ; B=B ; L=L } ;
      p={ v1=v1 ; v2=v2 } } ;
}
```

Lexicon

In this example, functions are linearized by using two different operations defined for: regular inflection of verbs (used in *write_V2*), where the verb is given in perfect tense, third person, singular, masculine and irregular inflection of verbs (used in *pray_V*), where two additional strings are given, namely the imperative singular and the imperative plural forms of the verb.

```
write_V2 = mkVerb "kiteb" ;
pray_V = mkVerb "talab" "itlob" "itolbu";
```

4.4. Inflection paradigm

An example of the output produced by GF for the verb 'write' is illustrated in Table 1.

Hebrew	Maltese
mkVPaal "ktb"	mkVerb "kiteb"
Perfect	
Vp1Sg ⇒ "ktbty"	(Per1 Sg) ⇒ "ktibt"
Vp1Pl ⇒ "ktbnw"	(Per1 Pl) ⇒"ktibna"
Vp2SgMasc ⇒ "ktbt"	(Per2 Sg) ⇒"ktibt"
Vp2SgFem ⇒ "ktbt"	
Vp2PlMasc ⇒ "ktbtM"	(Per2 Pl) ⇒"ktibtu"
Vp2PlFem ⇒ "ktbtN"	
Vp3SgMasc ⇒ "ktb"	(Per3Sg Masc) ⇒ "kiteb"
Vp3SgFem ⇒ "ktbh"	(Per3Sg Fem) ⇒ "kitbet"
Vp3PlMasc ⇒ "ktbw"	Per3Pl ⇒ "kitbu"
Vp3PlFem ⇒ "ktbw"	
Imperfect	
Vp1Sg ⇒ "Aktwb"	(Per1 Sg) ⇒ "nikteb"
Vp1Pl ⇒ "nktwb"	(Per1 Pl) ⇒ "niktbu"
Vp2SgMasc ⇒ "tktwb"	(Per2 Sg) ⇒ "tikteb"
Vp2SgFem ⇒ "tktby"	
Vp2PlMasc ⇒ "tktbw"	(Per2 Pl) ⇒ "tiktbu"
Vp2PlFem ⇒ "tktbw"	
Vp3SgMasc ⇒ "yktwb"	(Per3Sg Masc) ⇒ "jikteb"
Vp3SgFem ⇒ "tktwb"	(Per3Sg Fem) ⇒ "tikteb"
Vp3PlMasc ⇒ "yktbw"	Per3Pl ⇒ "jiktbu"
Vp3PlFem ⇒ "yktbw"	

Table 1: Example of Hebrew and Maltese verb inflection tables of the verb 'write'.

5. State of the work

The core syntax implemented for the two languages has around 13 categories and 22 construction functions. It covers simple syntactic constructions including predication rules which are built from noun and verb phrases.

The lexicons were manually populated with a small number of lexical units, covering around 20 verbs and 10 nouns in each language. The Maltese verb morphology covers the root groups: strong, defective and quadriliteral. In Hebrew, the strong verb paradigms and five weak verb paradigms in binyan pa'al are covered.

6. Discussion and related work

Although there are already some morphological analyzers available for Hebrew (Itai and Wintner, 2008; Yona and Wintner., 2008) and data resources available for Maltese (Rosner et al., 1999), they are not directly usable within the Grammatical Framework. To exploit the advantages offered by GF, the language's grammar must be implemented in this formalism. One of the advantages of implementing Semitic non-concatenative morphology in a typed language such as GF compared with other finite state languages is that strings are formed by records, and not through concatenation. Moreover, once the core grammar is defined and the structure and the form of the lexicon is determined, it is possible to automatically acquire lexical entries from exiting lexical resources. In the context of GF, three wide-coverage lexicons have been acquired automatically: Bulgarian (Angelov, 2008b), Finnish (Tutkimuskeskus, 2006) and Swedish (Angelov, 2008a).

In this work, the design decisions taken by the programmers are based on different points of arguments concerning the division of labour between a linguistically trained grammarian and a lexicographer. The Maltese implementation consider stems in the lexicon rather than patterns and roots, cf. Rosner et al. (1998); in the framework of GF, classes of inflectional phenomena are given an abstract representation that interact with the root and pattern system. In Hebrew, recognizing prefixes and suffixes is not always sufficient for recognizing the root of the verb. Although root recognition is mandatory for generating the verb's complete conjugation table, changes in patterns and the absence of root letters in different lexemes make it increasingly hard to infer the root (Deutsch and Frost, 2002) which requires a large amount of tri-consonantal constraints. This is in particular true for lexemes derived from weak roots where one of the root consonants is often missing (Frost et al., 2000). To avoid a large amount of morphosyntactic rules, we choose to employ semantic markings in the lexicon by specifying roots and patterns instead of lexemes; this computationally motivated approach becomes plausible since the meaning of the lexeme is already known.

7. Conclusions and Future Work

In this paper we have presented implementations of Hebrew and Maltese components that tend to convey the non-concatenative morphology of their verbs. Although we could identify common characteristics among these two Semitic languages, we found it difficult to generalize morphosyntactic rules across Semitic verbs when the focus is towards a computational motivated lexicon.

When designing a computer system that can process several languages automatically it is useful to generalize as many morphosyntactic rules across languages that belong to the same language group. One fundamental question that rises from our implementations is to what extent we can generalize the concrete syntaxes of Semitic languages. One way to approach this question is by employing semantic markings in the lexicons of the Semitic languages and focus on semantic aspects of morphological processing. This remains for future work.

8. References

Krasimir Angelov. 2008a. Importing SALDO in GF. http://spraakbanken.gu.se/personal/lars/kurs/lgres08/tpaper/angelov.pdf.

Krasimir Angelov. 2008b. Type-theoretical Bulgarian grammar. In In B. Nordström and eds. A. Ranta, editors, *Advances in Natural Language Processing (GoTAL 2008)*, volume 5221 of *LNCS/LNAI*, pages 52—-64.

Joseph Aquilina. 1960. *The structure of Maltese*. Valletta: Royal University of Malta.

Joseph Aquilina. 1962. *Papers in Maltese linguistics*. Royal University, Malta.

Maya Arad. 2005. *Roots and patterns: Hebrew morpho-syntax*. Springer, The Netherlands.

Edna Amir Coffin and Shmuel Bolozky. 2005. *A Reference Grammar of Modern Hebrew*. Cambridge University Press.

Ali Dada and Aarne Ranta. 2007. Implementing an open source arabic resource grammar in GF. In M. Mughazy, editor, *Perspectives on Arabic Linguistics.*, Papers from the Twentieth Annual Symposium on Arabic Linguistics. John Benjamins Publishing Company, March 26.

Avital Deutsch and Ram Frost, 2002. *Lexical organization and lexical access in a non-concatenated morphology*, chapter 9. John Benjamins.

R. Frost, A. Deutsch, and K. I. Forster. 2000. Decomposing morphologically complex words in a nonlinear morphology. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 26(3):751–765.

Gideon Goldberg. 1994. Principles of Semitic word-structure. In G. Goldberg and S. Raz, editors, *In Semantic and ushitic studies*, pages 29–64. Wiesbaden:Harrassowitz.

Robert D. Hoberman and M. Aronoff, 2003. *The verbal morphology of Maltese: From Semitic to Romance*, chapter 3, pages 61–78. Amsterdam: John Benjamins.

Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March.

P. Martin-Löf. 1975. An intuitionistic theory of types: Predicative part. In H. E. Rose and J. C. Shepherdson, editors, *Proc. of Logic Colloquium '73, Bristol, UK*, volume 80, pages 73–118. North-Holland.

John J. McCarthy. 1979. On stress and syllabification. *Linguistic Inquiry*, 10:443–465.

John J. McCarthy. 1981. The representation of consonant length in Hebrew. *Linguistic Inquiry*, 12:322–327.

Manwel Mifsud. 1995. *Loan verbs in Maltese: a descriptive and comparative study*. Leiden, New York, USA.

Aarne Ranta. 2004. Grammatical framework, a type-theoretical grammar formalism. *Journal of Functional Programming*, 14(2):145–189.

Aarne Ranta. 2009. The GF resource grammar library. *The on-line journal Linguistics in Language Technology (LiLT)*, 2(2). http://elanguage.net/journals/index.php/lilt/article/viewFile/214/158.

M. Rosner, J. Caruana R., and Fabri. 1998. Maltilex: a computational lexicon for Maltese. In *Semitic '98: Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 97–101, Morristown, NJ, USA. Association for Computational Linguistics.

M. Rosner, J. Caruana, and R. Fabri. 1999. Linguistic and computational aspects of maltilex. In *Arabic Translation and Localisation Symposium: Proceedings of the Workshop*, pages 2—-10, Tunis.

Kotimaisten Kielten Tutkimuskeskus. 2006. KOTUS wordlist. http://kaino.kotus.fi/sanat/nykysuomi.

Adam Ussishkin and Alina Twist. 2007. Lexical access in Maltese using visual and auditory lexical decision. In *Conference of Maltese Linguistics*, L-Ghaqda Internazzjonali tal-Lingwistika Maltija, University of Bremen, Germany, October.

Shlomo Yona and Shuly Wintner. 2008. A finite-state morphological grammar of Hebrew. *Natural Language Engineering*, 14(2):173–190, April.

Syllable Based Transcription of English Words into Perso-Arabic Writing System

Jalal Maleki
Dept. of Computer and Information Science
Linkping University
SE-581 83 Linkping
Sweden
Email: jma@ida.liu.se

Abstract

This paper presents a rule-based method for transcription of English words into the Perso-Arabic orthography. The method relies on the phonetic representation of English words such as the CMU pronunciation dictionary. Some of the challenging problems are the context-based vowel representation in the Perso-Arabic writing system and the mismatch between the syllabic structures of English and Persian. With some minor extensions, the method can be applied to English to Arabic transliteration as well.

1 Introduction

During the translation process from English to Persian certain words (usually names and trademarks) are transcribed rather than translated. This is a general issue in machine translation between language pairs. Unfortunately, there are no guidelines as to how these words should be written in the Perso-Arabic Script (PA-Script) and some words are written in more than 10 different ways ([9]). This paper introduces a rule-base method for English to PA-Script transcription which is based on the syllable structure of words. Syllables are important since transcription of vowels is mainly determined by the structure of the syllable in which the vowel appears. Given an English word we use a syllabified version of the CMU pronunciation dictionary (CMUPD) to lookup its pronunciation and use it for generating a phonemic romanized Persian transcription of the word which is finally resyllabified and transcribed into the Perso-Arabic Script (PA-Script) according to the syllabification-based method described in [11]. The romanized scheme we use is the Dabire-romanization described in [10]. Since Arabic and Persian essentially use the same script and have the same syllabic structure, our method can easily be extended to the Arabic script.

2 Phonological Issues

The essence of our method is phonological mapping between English and Persian and is defined as phonemic mapping of consonents and vowels and resyllabification of the source word using Persian syllable constraints. Just like transliteration between Arabic and English ([2]), transcription between English and Persian is a dfficult task. However, although the mapping between the sounds of Persian and english consonants and vowels is non-trivial, the most complicated step is conversion of Persian vowels to PA-Script [11].

2.1 Consonants

Mapping English consonants into Persian phonology is imperfect but straightforward and it can be summarized as a lookup operation. The mapping is however not perfect and in many cases a consonant is mapped into a Persian consonant that only approximately reflects its original pronunciation. For example, /th/ in 'thanks' (/TH, AE1, NG, K, S/) is transcribed to /t/, whereas, the /th/ of 'that' (/DH, AE1, T/) is transcribed to Persian /d/.

2.2 Vowels

From a transcription point of view, vowel correspondence between Persian and English phonology is also imperfect and relatively simple. Some examples are shown in Table-1. Some English diphthongs are treated as two separate vowels whereas some others are interpreted as a single vowel.

Phonological mapping is followed by conversion of phonemic romanized Persian to PA-Script. Type of syllable containing a vowel and the characteristics of the neighboring graphemes determine the choice of grapheme (or allographs) for the vowel. As an example, Table-2 shows the various and digraphs used for writing the vowel /i/ in different contexts [11].

2.3 Syllable Constraints and Consonant Clusters

Syllable structure in Persian is restricted to (C)V(C)(C), whereas, English allows the more complex structure (C)(C)(C)V(C)(C)(C)(C).

One of the main problems in writing English words in PA-Script is the transformation of syllables. For example, the word 'question' represented as /K, W, EH1, S, CH, AH0, N/ in CMUPD with the syllables /K, W, EH1, S/ and /CH, AH0, N/ is transcribed to *kuesšen* one syllable at a time and finally resyllabified as *kues-šen* and transliterated to PA-Script کوئسشن. Resyllabification is necessary since consonant clusters are broken by vowel epenthesis.

In general, the Persian transcription of English words involves short vowel insertion into consonant clusters and resyllabification (See Table-3 for examples.)

3 The Implementation

Transcription of an English word *w* into P-Script involves a number of steps which are briefly discussed below.

1. *w* is looked up in the syllabified CMUPD dictionary [4] and its syllabified pronunciation $p(w)$ is retrieved. For example, given the word 'surgical', we get: ((S ER1) (JH IH0) (K AH0 L))

2. Syllables of $p(w)$ are transcribed to Dabire which is a phonemic orthorgraphy for Persian. For the 'surgical', we get *((s e r) (g i) (kâl))*.

3. The syllables are individually modified to fulfill the contraints of Persian syllable structures. For example, *spring* (CCCVCC) is transformed to *espering* (VCCVCVCC) using *e* epenthesis, `prompt` (CCVCCC) is transformed to *perompet* (CVCVCCVC). See Table-3 for more examples.

4. The resulting Dabire word is resyllabified. For example, *espering* is syllabified as *es.pe.ring*

5. Application of context-dependent replace rules [3] to enforce orthographical conventions of Persian [5, 13, 1]

6. Finally, the Dabire-word is transliterated to Perso-Arabic Unicode.

Step 1-3 are currently implemented in Lisp and steps 4-6 are implemented as transducers in XFST [3]

The syllabification step (4) which is one of the main modules of the system is explained further. The syllabification transducer works from left to right on the input string and ensures that the number of consonants in the onset is maximized. Given the syllabic structure of Persian, this essentially means that if a vowel, V, is preceded by a consonant, C, then CV initiates a syllable. For example, for a word such as *jârue*, the syllabification *jâ.ru.e* (CV.CV.V) is selected and *jâr.u.e* (CVC.V.V) is rejected. The correct syllabification would naturally lead to correct writing since as mentioned earlier, vowels are written differently depending on their position in the syllable.

The following XFST-definitions form the core of the syllabification [11]:

```
define Sy V|VC|VCC|CV|CVC|CVCC;

define Sfy C* V C* @->
              ... "." ||  _ Sy;
```

The first statement defines a language (`Sy`) containing all syllables of Dabire. `V`, `VC` etc. are defined as regular languages that represent well-formed syllables in Dabire. For example, CVCC is defined as,

```
define CVCC [C V C C] .o. ~$NotAllowed;
```

which defines the language containing all possible CVCC syllables and excluding the untolerated consonant clusters in NotAllowed such as *bp*, *kq*, and *cc*.

Vowel	Example Word	Phonemes	Persian Phoneme	Romanized Persian	Perso-Arabic
AA	odd	AA D	â	âd	آد
AE	at	AE T	a	at	ات
AH	hut	HH AH T	â	hât	هات
AO	ought	AO T	o	ot	اوت
AW	cow	K AW	â	kâv	کاو
AY	hide	HH AY D	ây	hâyd	هاید

Table 1. *Some Vowels from CMU Pronunciation Dictionary with Examples*

The second statement defines a replacement rule [3] that represents the syllabification process. The operator `@>` ensures that the shortest possible strings (of the form `C* V C*`) are selected in left to right direction and identified as syllables which are separated by a dot.

Table-4 includes examples that illustrate examples of input/output for this.

4 Discussion and Evaluation

We have introduced a rule based transcription of English to PA-Script. Earlier work [2, 8, 6, 7] mainly relies on statistical methods.

Our method produces correct transcriptions for most of the data-set randomly selected from CMUPD. Quantitative evaluation of the method is in progress. The performance of the system is dependent on the availability of syllabified English words and future improvements would require use of statistical methods for automatically handling words that do not exist in the dictionary. Some early experiments [14] based on CMUPD show a success rate of 71.6% in automatic grapheme to phoneme conversion of English words not present in CMUPD. Further development would also require integration of automatic syllabification of English [12] into the system.

References

[1] M. S. Adib-Soltâni. *An Introduction to Persian Orthography - (in Persian)*. Amir Kabir Publishing House, Tehrân, 2000.

[2] Y. Al-Onaizan and K. Knight. Machine transliteration of names in arabic text. In *ACL Workshop on Computational Approaches to Semitic Languages*, 2002.

[3] K. R. Beesley and L. Karttunen. *Finite State Morphology*. CSLI Publications, 2003.

[4] Carnegie Mellon University. CMU pronunciation dictionary. *http://www.speech.cs.cmu.edu/cgi-bin/cmudict*, 2008.

[5] Farhangestan. *Dastur e Khatt e Farsi (Persian Orthography)*, volume Supplement No. 7. Persian Academy, Tehran, 2003.

[6] J. Johanson. Transcription of names written in farsi into english. In A. Farghaly and K. Megerdoomian, editors, *Proceedings of the 2nd workshop on computational approaches to Arabic Script-based languages*, pages 74–80, 2007.

[7] S. Karimi, A. Turpin, and F. Scholer. English to persian transliteration. In *Lecture Notes in Computer Science*, volume 4209, pages 255–266. Springer, 2006.

[8] M. M. Kashani, F. Popowich, and A. Sarkar. Automatic transliteration of proper nouns from arabic to english. In A. Farghaly and K. Megerdoomian, editors, *Proceedings of the 2nd workshop on computational approaches to Arabic Script-based languages*, pages 81–87, 2007.

[9] R. R. Z. Malek. *Qavâed e Emlâ ye Fârsi*. Golâb, 2001.

[10] J. Maleki. A Romanized Transcription for Persian. In *Proceedings of Natural Language Processing Track (INFOS2008), Cairo*, 2008.

[11] J. Maleki and L. Ahrenberg. Converting Romanized Persian to Arabic Writing System Using Syllabification. In *Proceedings of the LREC2008, Marrakech*, 2008.

[12] Y. Marchand, C. R. Adsett, and R. I. Damper. Automatic Syllabification in English: A Comparison of Different Algorithms. *Language and Speech*, 52(1):1–27, 2009.

[13] S. Neysari. *A Study on Persian Orthography - (in Persian)*. Sâzmân e Câp o Entešârât, 1996.

[14] S. Stymne. Private communication. *Linköping*, 2010.

/i/	Word Initial	Segment Initial	Segment Medial	Segment Final	Intra-Word Isolated
V, VC, VCC	ایـ این	ئیـ پائیز	ـئیـ لئیم	ـئی خالیئی	ـای ,ـئی رفته‌ای ,بانوئی
CVC, CVCC		ـیـ پردیس	ـیـ سیزده		
CV		ـیـ دیدار	ـیـ بیدار	ـی خاکی	ـی کاری

Table 2. *Mapping /i/ to P-Script Graphemes*

English	Onset/Coda	Transcription	Example	Clusters
/šr/	Onset	/šer/	shrink→šerink	/šr/
/sC_1/	Onset	/esC_1/	school→eskul	/sp, st, sk, sm, sn, sl/
/šC_2/	Onset	/ešC_2/	schmock→ešmâk	/šp, št, šk, šm, šn, šl/
/C_3C_1/	Onset	/C_3eC_1/	trunk→terânk	/pr, pl, bl, br, .../
/sCw/	Onset	/esCu/	squash→eskuâš	/skw/
/sCy/	Onset	/esCiy/	student→estiyudent	/spy, sty/
/sCC_1/	Onset	/esCeC_1/	spring→espering	/spl, spr, str, skr/
/C_1Cs/	Coda	/C_1Ces/	corps→korpes	/lps, rps, rts, rks/
/CCCC/	Coda	/CCeCeC/	prompts→perâmpetes	

Table 3. *Epenthesis in consonant cluster transcription. C_1 stands for all consonants except /w/ and /y/. C_2 stands for all consonants except /w/, /y/ and /r/. C_3 Stands for all consonants except /s/ and /š/.*

English Word	CMU Pronunciation	Dabire Romanization	Syllabification	PA-Script
GEORGE	JH AO1 R JH	jorj	jorj	جورج
BUSH	B UH1 SH	buš	buš	بوش
BIOGEN	B AY1 OW0 JH EH2 N	bâyojen	bâ.yo.jen	بایوجن
LOUISE	L UW0 IY1 Z	luiz	lu.iz	لوئیز
LOUISIANA	L UW0 IY2 Z IY0 AE1 N AH0	luizianâ	lu.i.zi.a.nâ	لوئیزیئنا
INDOSUEZ	IH1 N D OW0 S UW0 EY1 Z	indosuez	in.do.su.ez	ایندو سوئز
SPRITE	S P R AY1 T	esperâyt	es.pe.râyt	اسپرایت

Table 4. *Examples showing some of the steps in the transliteration*

COLABA: Arabic Dialect Annotation and Processing

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, Yassine Benajiba

Center for Computational Learning Systems
475 Riverside Drive, Suite 850
New York, NY 10115
Columbia University
{mdiab,habash,rambow,mtantawy,ybenajiba}@ccls.columbia.edu

Abstract

In this paper, we describe COLABA, a large effort to create resources and processing tools for Dialectal Arabic Blogs. We describe the objectives of the project, the process flow and the interaction between the different components. We briefly describe the manual annotation effort and the resources created. Finally, we sketch how these resources and tools are put together to create DIRA, a term-expansion tool for information retrieval over dialectal Arabic collections using Modern Standard Arabic queries.

1. Introduction

The Arabic language is a collection of historically related variants. Arabic dialects, collectively henceforth Dialectal Arabic (DA), are the day to day vernaculars spoken in the Arab world. They live side by side with Modern Standard Arabic (MSA). As spoken varieties of Arabic, they differ from MSA on all levels of linguistic representation, from phonology, morphology and lexicon to syntax, semantics, and pragmatic language use. The most extreme differences are on phonological and morphological levels.

The language of education in the Arab world is MSA. DA is perceived as a lower form of expression in the Arab world; and therefore, not granted the status of MSA, which has implications on the way DA is used in daily written venues. On the other hand, being the spoken language, the native tongue of millions, DA has earned the status of living languages in linguistic studies, thus we see the emergence of serious efforts to study the patterns and regularities in these linguistic varieties of Arabic (Brustad, 2000; Holes, 2004; Bateson, 1967; Erwin, 1963; Cowell, 1964; Rice and Sa'id, 1979; Abdel-Massih et al., 1979). To date most of these studies have been field studies or theoretical in nature with limited annotated data. In current statistical Natural Language Processing (NLP) there is an inherent need for large-scale annotated resources for a language. For DA, there has been some limited focused efforts (Kilany et al., 2002; Maamouri et al., 2004; Maamouri et al., 2006); however, overall, the absence of large annotated resources continues to create a pronounced bottleneck for processing and building robust tools and applications.

DA is a pervasive form of the Arabic language, especially given the ubiquity of the web. DA is emerging as the language of informal communication online, in emails, blogs, discussion forums, chats, SMS, etc, as they are media that are closer to the spoken form of language. These genres pose significant challenges to NLP in general for any language including English. The challenge arises from the fact that the language is less controlled and more speech like while many of the textually oriented NLP techniques are tailored to processing edited text. The problem is compounded for Arabic precisely because of the use of DA in these genres. In fact, applying NLP tools designed for MSA directly to DA yields significantly lower performance, making it imperative to direct the research to building resources and dedicated tools for DA processing.

DA lacks large amounts of consistent data due to two factors: a lack of orthographic standards for the dialects, and a lack of overall Arabic content on the web, let alone DA content. These lead to a severe deficiency in the availability of computational annotations for DA data. The project presented here – Cross Lingual Arabic Blog Alerts (COLABA) – aims at addressing some of these gaps by building large-scale annotated DA resources as well as DA processing tools.[1]

This paper is organized as follows. Section 2. gives a high level description of the COLABA project and reviews the project objectives. Section 3. discusses the annotated resources being created. Section 4. reviews the tools created for the annotation process as well as for the processing of the content of the DA data. Finally, Section 5. showcases how we are synthesizing the resources and tools created for DA for one targeted application.

2. The COLABA Project

COLABA is a multi-site partnership project. This paper, however, focuses only on the Columbia University contributions to the overall project.

COLABA is an initiative to process Arabic social media data such as blogs, discussion forums, chats, etc. Given that the language of such social media is typically DA, one of the main objective of COLABA is to illustrate the significant impact of the use of dedicated resources for the processing of DA on NLP applications. Accordingly, together with our partners on COLABA, we chose Information Retrieval (IR) as the main testbed application for our ability to process DA.

Given a query in MSA, using the resources and processes created under the COLABA project, the IR system is able to retrieve relevant DA blog data in addition to MSA data/blogs, thus allowing the user access to as much Arabic

[1] We do not address the issue of augmenting Arabic web content in this work.

content (in the inclusive sense of MSA and DA) as possible. The IR system may be viewed as a cross lingual/cross dialectal IR system due to the significant linguistic differences between the dialects and MSA. We do not describe the details of the IR system or evaluate it here; although we allude to it throughout the paper.

There are several crucial components needed in order for this objective to be realized. The COLABA IR system should be able to take an MSA query and convert it/translate it, or its component words to DA or alternatively convert all DA documents in the search collection to MSA before searching on them with the MSA query. In COLABA, we resort to the first solution. Namely, given MSA query terms, we process them and convert them to DA. This is performed using our DIRA system described in Section 5.. DIRA takes in an MSA query term(s) and translates it/(them) to their corresponding equivalent DA terms. In order for DIRA to perform such an operation it requires two resources: a lexicon of MSA-DA term correspondences, and a robust morphological analyzer/generator that can handle the different varieties of Arabic. The process of creating the needed lexicon of term correspondences is described in detail in Section 3.. The morphological analyzer/generator, MAGEAD, is described in detail in Section 4.3..

For evaluation, we need to harvest large amounts of data from the web. We create sets of queries in domains of interest and dialects of interest to COLABA. The URLs generally serve as good indicators of the dialect of a website; however, given the fluidity of the content and variety in dialectal usage in different social media, we decided to perform dialect identification on the lexical level.

Moreover, knowing the dialect of the lexical items in a document helps narrow down the search space in the underlying lexica for the morphological analyzer/generator. Accordingly, we will also describe the process of dialect annotation for the data.

The current focus of the project is on blogs spanning four different dialects: Egyptian (EGY), Iraqi (IRQ), Levantine (LEV), and (a much smaller effort on) Moroccan (MOR). Our focus has been on harvesting blogs covering 3 domains: social issues, religion and politics.

Once the web blog data is harvested as described in Section 3.1., it is subjected to several processes before it is ready to be used with our tools, namely MAGEAD and DIRA. The annotation steps are as follows:

1. **Meta-linguistic Clean Up.** The raw data is cleaned from html mark up, advertisements, spam, encoding issues, and so on. Meta-linguistic information such as date and time of post, poster identity information and such is preserved for use in later stages.

2. **Initial Ranking of the Blogs.** The sheer amount of data harvested is huge; therefore, we need to select blogs that have the most dialectal content so as to maximally address the gap between MSA and DA resources. To that end, we apply a simple DA identification (DI) pipeline to the blog document collection ranking them by the level of dialectal content. The DI pipeline is described in detail in Section 4.2.. The intuition is that the more words in the blogs that are not analyzed or recognized by a MSA morphological analyzer, the more dialectal the blog. It is worth noting that at this point we only identify that words are not MSA and we make the simplifying assumption that they are DA. This process results in an initial ranking of the blog data in terms of dialectness.

3. **Content Clean-Up.** The content of the highly ranked dialectal blogs is sent for an initial round of manual clean up handling speech effects and typographical errors (typos) (see Section3.2.). Additionally, one of the challenging aspects of processing blog data is the severe lack of punctuation. Hence, we add a step for sentence boundary insertion as part of the cleaning up process (see Section 3.3.). The full guidelines will be presented in a future publication.

4. **Second Ranking of Blogs and Dialectalness Detection**. The resulting cleaned up blogs are passed through the DI pipeline again. However, this time, we need to identify the actual lexical items and add them to our lexical resources with their relevant information. In this stage, in addition to identifying the dialectal unigrams using the DI pipeline as described in step 2, we identify out of vocabulary bigrams and trigrams allowing us to add entries to our created resources for words that look like MSA words (i.e. cognates and faux amis that already exist in our lexica, yet are specified only as MSA). This process renders a second ranking for the blog documents and allows us to hone in on the most dialectal words in an efficient manner. This process is further elaborated in Section 4.2..

5. **Content Annotation**. The content of the blogs that are most dialectal are sent for further content annotation. The highest ranking blogs undergo full word-by-word dialect annotation as described in Section 3.5.. Based on step 4, the most frequent surface words that are deemed dialectal are added to our underlying lexical resources. Adding an entry to our resources entails rendering it in its lemma form since our lexical database uses lemmas as its entry forms. We create the underlying lemma (process described in Section 3.6.) and its associated morphological details as described in Section 3.7.. Crucially, we tailor the morphological information to the needs of MAGEAD. The choice of surface words to be annotated is ranked based on the word's frequency and its absence from the MSA resources. Hence the surface forms are ranked as follows: unknown frequent words, unknown words, then known words that participate in infrequent bigrams/trigrams compared to MSA bigrams/trigrams. All the DA data is rendered into a Colaba Conventional Orthography (CCO) described in Section 3.4.. Annotators are required to use the CCO for all their content annotations.

To efficiently clean up the harvested data and annotate its content, we needed to create an easy to use user interface

with an underlying complex database repository that organizes the data and makes it readily available for further research. The annotation tool is described in Section 4.1..

3. Resource Creation

Resource creation for COLABA is semi automatic. As mentioned earlier, there is a need for a large collection of data to test out the COLABA IR system. The data would ideally have a large collection of blogs in the different relevant dialects in the domains of interest, annotated with the relevant levels of linguistic knowledge such as degree of dialectness and a lexicon that has coverage of the lexical items in the collection. Accordingly, the blog data is harvested using a set of identified URLs as well as queries that are geared towards the domains of interest in the dialect.

3.1. Data Harvesting

Apart from identifying a set of URLs in each of the relevant dialects, we designed a set of DA queries per dialect to harvest large quantities of DA data from the web. These queries were generated by our annotators with no restrictions on orthographies, in fact, we gave the explicit request that they provide multiple variable alternative orthographies where possible. The different dialects come with their unique challenges due to regional variations which impact the way people would orthographically represent different pronunciations. For example, DA words with MSA cognates whose written form contains the ق q^2 (*Qaf*) consonant may be spelled etymologically (as ق *q*) or phonologically as one of many local variants: ك *k*, أ *Â* or گ *G*.
We collected 40 dialectal queries from each of our 25 annotators specifically asking them when possible to identify further regional variations. In our annotations in general, we make the gross simplifying assumption that Levantine (Syrian, Lebanese, Palestinian and Jordanian) Arabic is a single dialect. However, for the process of query generation, we asked annotators to identify sub-dialects. So some of our queries are explicitly marked as Levantine-Palestinian or Levantine-Syrian for instance. Moreover, we asked the annotators to provide queries that have verbs where possible. We also asked them to focus on queries related to the three domains of interest: politics, religion and social issues. All queries were generated in DA using Arabic script, bearing in mind the lack of orthographic standards. The annotators were also asked to provide an MSA translation equivalent for the query and an English translation equivalent. Table 1 illustrates some of the queries generated.

3.2. Typographical Clean Up

Blog data is known to be a challenging genre for any language from a textual NLP perspective since it is more akin to spoken language. Spelling errors in MSA (when used) abound in such genres which include speech effects. The problem is compounded for Arabic since there are no DA orthographic standards. Accordingly, devising guidelines for such a task is not straight forward. Thus, we simplified the task to the narrow identification of the following categories:

- MSA with non-standard orthography, e.g., هذة *hðħ* 'this' becomes هذه *hðh*, and المساجذ *AlmsAjð* 'mosques' becomes المساجد *AlmsAjd*.
- Speech Effects (SE) are typical elongation we see in blog data used for emphasis such as كووووره *kwwwwrh* 'ball' is rendered كورة *kwrħ*.
- Missing/Added Spaces (MS/AS) are cases where there is obviously a missing space between two or more words that should have been rendered with a space. For example, in EGY, متكلشالبرتأنة *mtklšAlbrtÂnħ* 'don't eat the orange' is turned into متكلش البرتأنة *mtklš AlbrtÂnħ*. Note that in this dialectal example, we do not require the annotator to render the word for orange البرتأنة *AlbrtÂnħ* in its MSA form, namely, البرتقالة *AlbrtqAlħ*.

3.3. Sentence Boundary Detection

In blogs, sentence boundaries are often not marked explicitly with punctuation. In this task, annotators are required to insert boundaries between sentences. We define a sentence in our guidelines as a syntactically and semantically coherent unit in language. Every sentence has to have at least a main predicate that makes up a main clause. The predicate could be a verb, or in the case of verb-less sentences, the predicate could be a nominal, adjectival or a prepositional phrase. Table 2 illustrates a blog excerpt as it occurs naturally on the web followed by sentence boundaries explicitly inserted with a carriage return splitting the line in three sentences.

3.4. COLABA Conventional Orthography

Orthography is a way of writing language using letters and symbols. MSA has a standard orthography using the Arabic script. Arabic dialects, on the other hand, do not have a standard orthographic system. As such, a variety of approximations (phonological/lexical/etymological) are often pursued; and they are applied using Arabic script as well as Roman/other scripts. In an attempt to conventionalize the orthography, we define a phonological scheme which we refer to as the COLABA Conventional Orthography (CCO). This convention is faithful to the dialectal pronunciation as much as possible regardless of the way a word is typically written. This scheme preserves and explicitly represents all the sounds in the word including the vowels. For example, باب *bAb* 'door' is rendered as *be:b* in CCO for LEV (specifically Lebanese) but as *ba:b* for EGY.[3] The full guidelines will be detailed in a future publication.

[2] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter (HSB) transliteration scheme (Habash et al., 2007).

[3] Most CCO symbols have English-like/HSB-like values, e.g., ب *b* or م *m*. Exceptions include *T* (ث *θ*), *D* (ذ ð), *c* (ش š), *R* (غ *γ*), *7* (ح *H*), *3* (ع ʕ), and *2* (ء ʾ). CCO uses '.' to indicate emphasis/velarization, e.g., *t.* (ط *T*).

DA Query	DA	MSA	English
الطلاق بقى ظاهره	EGY	الطلاق اصبح ظاهرة	divorce became very common
راح احچيلكم	IRQ	سوف اروي لكم	I will tell you a story
راح دوز ع قبر بيه	LEV	ذهب فورا الى قبر ابيه	He went directly to visit his father's tomb
ما زال شاد فراسو	MOR	لا زال في وضع جيد	he is still in good shape

Table 1: Sample DA queries used for harvesting blog data

Input text
بدي اخذ ماجيستير ودكتوراه بدي اتزوج وجيب ولاد وبدي عيش بحو اسري كله مودة
After manual sentence boundary detection
بدي اخذ ماجيستير ودكتوراه بدي اتزوج وجيب ولاد وبدي عيش بحو اسري كله مودة

Table 2: LEV blog excerpt with sentence boundaries identified.

- CCO explicitly indicates the pronounced short vowels and consonant doubling, which are expressed in Arabic script with optional diacritics. Accordingly, there is no explicit marking for the sukuun diacritic which we find in Arabic script. For example, the CCO for مركب *mrkb* in EGY could be *markib* 'boat' or *mirakkib* 'something put together/causing to ride' or *murakkab* 'complex'.
- Clitic boundaries are marked with a +. This is an attempt at bridging the gap between phonology and morphology. We consider the following affixations as clitics: conjunctions, prepositions, future particles, progressive particles, negative particles, definite articles, negative circumfixes, and attached pronouns. For example, in EGY CCO وسلام *wslAm* 'and peace' is rendered *we+sala:m* and مايكتبش *mAyktbš* 'he doesn't write' is rendered *ma+yiktib+c*.
- We use the ^ symbol to indicate the presence of the Ta Marbuta (feminine marker) morpheme or of the Tanween (nunation) morpheme (marker of indefiniteness). For example, مكتبة *mktbħ* 'library' is rendered in CCO as *maktaba^* (EGY). Another example is عملياً *ʕmlyAã* 'practically', which is rendered in CCO as *3amaliyyan^*.

CCO is comparable to previous efforts on creating resources for Arabic dialects (Maamouri et al., 2004; Kilany et al., 2002). However, unlike Maamouri et al. (2004), CCO is not defined as an Arabic script dialectal orthography. CCO is in the middle between the morphophonemic and phonetic representations used in Kilany et al. (2002) for Egyptian Arabic. CCO is quite different from commonly used transliteration schemes for Arabic in NLP such as Buckwalter transliteration in that CCO (unlike Buckwalter) is not bijective with Arabic standard orthography.
For the rest of this section, we will use CCO in place of the HSB transliteration except when indicated.

3.5. Dialect Annotation

Our goal is to annotate all the words in running text with their degree of dialectalness. In our conception, for the purposes of COLABA we think of MSA as a variant dialect; hence, we take it to be the default case for the Arabic words in the blogs. We define a dialectal scale with respect to orthography, morphology and lexicon. We do not handle phrasal level or segment level annotation at this stage of our annotation, we strictly abide by a word level annotation.[4] The annotators are required to provide the CCO representation (in Section 3.4.) for all the words in the blog. If a word as it appears in the original blog maintains its meaning and orthography as in MSA then it is considered the default MSA for dialect annotation purposes, however if it is pronounced in its context dialectically then its CCO representation will reflect the dialectal pronunciation, e.g. يكتب, *yktb* 'he writes' is considered MSA from a dialect annotation perspective, but in an EGY context its CCO representation is rendered *yiktib* rather than the MSA CCO of *yaktub*.
Word dialectness is annotated according to a 5-point scale building on previous efforts by Habash et al. (2008):

- WL1: MSA with dialect morphology بيكتب *bi+yiktib* 'he is writing', هيكتب *ha+yiktib* 'he will write'
- WL2: MSA faux amis where the words look MSA but are semantically used dialectically such as عم *3am* a LEV progressive particle meaning 'in the state of' or MSA 'uncle'
- WL3: Dialect lexeme with MSA morphology such as سيزعل *sa+yiz3al* 'he will be upset'
- WL4: Dialect lexeme where the word is simply a dialectal word such as the negative particle مش *mic* 'not'

[4] Annotators are aware of multiword expressions and they note them when encountered.

- WL5: Dialect lexeme with a consistent systematic phonological variation from MSA, e.g., LEV تلاتة *tala:te^* 'three' versus ثلاثة *Tala:Ta^*.

In addition, we specify another six word categories that are of relevance to the annotation task on the word level: Foreign Word (جيلاتو, *jila:to*, 'gelato ice cream'), Borrowed Word (ويك اند, *wi:k 2end*, 'weekend'), Arabic Named Entity (عمرو دياب, *3amr dya:b*, 'Amr Diab'), Foreign Named Entity (جيمي كارتر, *jimi kartar*, 'Jimmy Carter'), Typo (further typographical errors that are not caught in the first round of manual clean-up), and in case they don't know the word, they are instructed to annotate it as unknown.

3.6. Lemma Creation

This task is performed for a subset of the words in the blogs. We focus our efforts first on the cases where an MSA morphological analyzer fails at rendering any analysis for a given word in a blog. We are aware that our sampling ignores the faux amis cases with MSA as described in Section 3.5.. Thus, for each chosen/sampled dialectal surface word used in an example usage from the blog, the annotator is required to provide a lemma, an MSA equivalent, an English equivalent, and a dialect ID. All the dialectal entries are expected to be entered in the CCO schema as defined in Section 3.4..

We define a lemma (citation form) as the basic entry form of a word into a lexical resource. The lemma represents the semantic core, the most important part of the word that carries its meaning. In case of nouns and adjectives, the lemma is the definite masculine singular form (without the explicit definite article). And in case of verbs, the lemma is the 3rd person masculine singular perfective active voice. All lemmas are clitic-free.

A dialectal surface word may have multiple underlying lemmas depending on the example usages we present to the annotators. For example, the word مركبه *mrkbh* occurs in two examples in our data: 1. سامي مركبه بإيديه *sa:mi mirakkib+uh be+2ide:+h* 'Sami built it with his own hands' has the corresponding EGY lemma *mirakkib* 'build'; and 2. الرجالة راحوا يشتروا مركبه منه *ir+rigga:la^ ra:7u yictiru markib+uh minn+uh* 'The men went to buy his boat from him' with the corresponding lemma *markib* 'boat'. The annotators are asked to explicitly associate each of the created lemmas with one or more of the presented corresponding usage examples.

3.7. Morphological Profile Creation

Finally, we further define a morphological profile for the entered lemmas created in Section 3.6.. A computationally oriented morphological profile is needed to complete the necessary tools relevant for the morphological analyzer MAGEAD (see Section 4.3.). We ask the annotators to select (they are given a list of choices) the relevant part-of-speech tag (POS) for a given lemma as it is used in the blogs. For some of the POS tags, the annotators are requested to provide further morphological specifications.

In our guidelines, we define coarse level POS tags by providing the annotators with detailed diagnostics on how to identify the various POS based on form, meaning, and grammatical function illustrated using numerous examples. The set of POS tags are as follows: (Common) Noun, Proper Noun, Adjective, Verb, Adverb, Pronoun, Preposition, Demonstrative, Interrogative, Number, and Quantifier. We require the annotators to provide a detailed morphological profile for three of the POS tags mentioned above: Verb, Noun and Adjective. For this task, our main goal is to identify irregular morphological behavior. They transcribe all their data entries in the CCO representation only as defined in Section 3.4.. We use the Arabic script below mainly for illustration in the following examples.

- **Verb Lemma:** In addition to the basic 3rd person masculine singular (3MS) active perfective form of the dialectal verb lemma, e.g., شرب *cirib* 'he drank' (EGY), the annotators are required to enter: (i) the 3MS active imperfective يشرب *yicrab*; (ii) the 3MS passive perfective is انشرب *incarab*; (iii) the 3MS passive imperfective ينشرب *yincirib*; and (iv) and the masculine singular imperative اشرب *icrab*.

- **Noun Lemma:** The annotators are required to enter the feminine singular form of the noun if available. They are explicitly asked not to veer too much away from the morphological form of the lemma, so for example, they are not supposed to put ست *sit* 'woman/lady' as the feminine form of راجل *ra:gil* 'man'. The annotators are asked to specify the rationality/humanness of the noun which interacts in Arabic with morphosyntactic agreement. Additional optional word forms to provide are any broken plurals, mass count plural collectives, and plurals of plurals, e.g *rigga:la^* and *riga:l* 'men' are both broken plurals of *ra:gil* 'man'.

- **Adjective Lemma:** For adjectives, the annotators provide the feminine singular form and any broken plurals, e.g. the adjective أول *2awwel* 'first [masc.sing]' has the feminine singular form أولى *2u:la* and the broken plural أوائل *2awa:2il*.

4. Tools for COLABA

In order to process and manage the large amounts of data at hand, we needed to create a set of tools to streamline the annotation process, prioritize the harvested data for manual annotation, then use the created resources for MAGEAD.

4.1. Annotation Interface

Our annotation interface serves as the portal which annotators use to annotate the data. It also serves as the repository for the data, the annotations and management of the annotators. The annotation interface application runs on a web server because it is the easiest and most efficient way to allow different annotators to work remotely, by entering their annotations into a central database. It also manages the annotators tasks and tracks their activities efficiently. For a more detailed description of the interface see (Benajiba and

Diab, 2010). For efficiency and security purposes, the annotation application uses two different servers. In the first one, we allocate all the html files and dynamic web pages. We use PHP to handle the dynamic part of the application which includes the interaction with the database. The second server is a database server that runs on PostgreSQL.[5] Our database comprises 22 relational databases that are categorized into tables for:

- Basic information that is necessary for different modules of the application. These tables are also significantly useful to ease the maintenance and update of the application.
- User permissions: We have various types of users with different permissions and associated privileges. These tables allow the application to easily check the permissions of a user for every possible action.
- Annotation information: This is the core table category of our database. Its tables save the annotation information entered by each annotator. They also save additional information such as the amount of time taken by an annotator to finish an annotation task.

For our application, we define three types of users, hence three views (see Figure 1):

1. *Annotator.* An Annotator can perform an annotation task, check the number of his/her completed annotations, and compare his/her speed and efficiency against other annotators. An annotator can only work on one dialect by definition since they are required to possess native knowledge it. An annotator might be involved in more than one annotation task.
2. *Lead Annotator.* A Lead annotator (i) manages the annotators' accounts, (ii) assigns a number of task units to the annotators, and, (iii) checks the speed and work quality of the annotators. Leads also do the tasks themselves creating a gold annotation for comparison purposes among the annotations carried out by the annotators. A lead is an expert in only one dialect and thus s/he can only intervene for the annotations related to that dialect.
3. *Administrator.* An Administrator (i) manages the Leads' accounts, (ii) manages the annotators' accounts, (iii) transfers the data from text files to the database, (iv) purges the annotated data from the data base to xml files, and (v) produces reports such as inter-annotator agreement statistics, number of blogs annotated, etc.

The website uses modern JavaScript libraries in order to provide highly dynamic graphical user interfaces (GUI). Such GUIs facilitate the annotator's job leading to significant gain in performance speed by (i) maximizing the number of annotations that can be performed by a mouse click rather than a keyboard entry and by (ii) using color coding for fast checks. Each of the GUIs which compose our web applications has been carefully checked to be consistent with the annotation guidelines.

[5]http://www.postgresql.org/

4.2. DA Identification Pipeline

We developed a simple module to determine the degree to which a text includes DA words. Specifically, given Arabic text as input, we were interested in determining how many words are not MSA. The main idea is to use an MSA morphological analyzer, Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004), to analyze the input text. If BAMA is able to generate a morphological analysis for an input word, then we consider that word MSA.
As a result, we have a conservative assessment of the dialectness of an input text. A major source of potential errors are names which are not in BAMA.
We assessed our pipeline on sample blog posts from our harvested data. In an EGY blog post[6] 19% of the word types failed BAMA analysis. These words are mainly DA words with few named entities. Similar experiments were conducted on IRQ,[7] LEV,[8] and MOR[9] blog posts yielding 13.5%, 8% and 26% of non-MSA word types, respectively. It is worth noting the high percentage of out of vocabulary words for the Moroccan thread compared to the other dialects. Also, by comparison, the low number of misses for Levantine. This may be attributed to the fact that BAMA covers some Levantine words due to the LDC's effort on the Levantine Treebank (Maamouri et al., 2006).
We further analyzed BAMA-missed word types from a 30K word blog collection. We took a sample of 100 words from the 2,036 missed words. We found that 35% are dialectal words and that 30% are named entities. The rest are MSA word that are handled by BAMA. We further analyzed two 100 string samples of least frequent bigrams and trigrams of word types (measured against an MSA language model) in the 30K word collection. We found that 50% of all bigrams and 25% of trigrams involved at least one dialectal word. The percentages of named entities for bigrams and trigrams in our sample sets are 19% and 43%, respectively.

4.3. MAGEAD

MAGEAD is a morphological analyzer and generator for the Arabic language family, by which we mean both MSA and DA. For a fuller discussion of MAGEAD (including an evaluation), see (Habash et al., 2005; Habash and Rambow, 2006; Altantawy et al., 2010). For an excellent discussion of related work, see (Al-Sughaiyer and Al-Kharashi, 2004). MAGEAD relates (bidirectionally) a lexeme and a set of linguistic features to a surface word form through a sequence of transformations. In a generation perspective, the features are translated to abstract morphemes which are then ordered, and expressed as concrete morphemes. The concrete templatic morphemes are interdigitated and affixes added, finally morphological and phonological rewrite rules are applied. In this section, we discuss our organization of linguistic knowledge, and give some examples; a more complete discussion of the organization of linguistic knowledge in MAGEAD can be found in (Habash et al., 2005).

[6]http://wanna-b-a-bride.blogspot.com/2009/09/blog-post_29.html
[7]http://archive.hawaaworld.com/showthread.php?t=606067&page=76
[8]http://www.shabablek.com/vb/t40156.html
[9]http://forum.oujdacity.net/topic-t5743.html

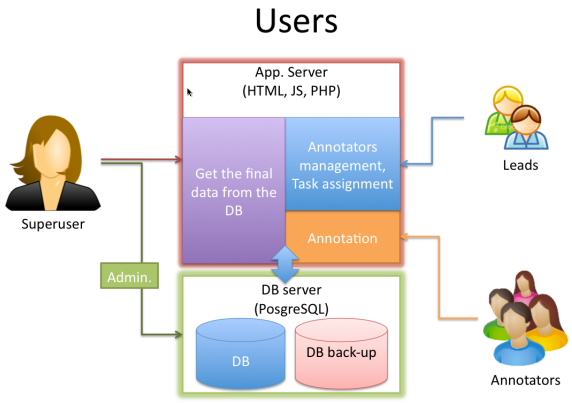

Figure 1: Servers and views organization.

Lexeme and Features Morphological analyses are represented in terms of a lexeme and features. We define the *lexeme* to be a triple consisting of a root, a *morphological behavior class* (MBC), and a meaning index. We do not deal with issues relating to word sense here and therefore do not further discuss the meaning index. It is through this view of the lexeme (which incorporates productive derivational morphology without making claims about semantic predictability) that we can have both a lexeme-based representation, and operate without a lexicon (as we may need to do when dealing with a dialect). In fact, because lexemes have internal structure, we can hypothesize lexemes on the fly without having to make wild guesses (we know the pattern, it is only the root that we are guessing). Our evaluation shows that this approach does not overgenerate. We use as our example the surface form ازدهرت *Aizdaharat* (*Azdhrt* without diacritics) "she/it flourished". The MAGEAD lexeme-and-features representation of this word form is as follows:

(1) Root:zhr MBC:verb-VIII POS:V PER:3 GEN:F NUM:SG ASPECT:PERF

Morphological Behavior Class An MBC maps sets of linguistic feature-value pairs to sets of abstract morphemes. For example, MBC verb-VIII maps the feature-value pair ASPECT:PERF to the abstract root morpheme [PAT_PV:VIII], which in MSA corresponds to the concrete root morpheme *V1tV2V3*, while the MBC verb-II maps ASPECT:PERF to the abstract root morpheme [PAT_PV:II], which in MSA corresponds to the concrete root morpheme *1V22V3*. We define MBCs using a hierarchical representation with non-monotonic inheritance. The hierarchy allows us to specify only once those feature-to-morpheme mappings for all MBCs which share them. For example, the root node of our MBC hierarchy is a word, and all Arabic words share certain mappings, such as that from the linguistic feature conj:w to the clitic *w+*. This means that all Arabic words can take a cliticized conjunction. Similarly, the object pronominal clitics are the same for all transitive verbs, no matter what their templatic pattern is. We have developed a specification language for expressing MBC hierarchies in a concise manner. Our hypothesis is that the MBC hierarchy is Arabic variant-independent, i.e. DA/MSA independent. Although as more Arabic variants are added, some modifications may be needed. Our current MBC hierarchy specification for both MSA and Levantine, which covers only the verbs, comprises 66 classes, of which 25 are abstract, i.e., only used for organizing the inheritance hierarchy and never instantiated in a lexeme.

MAGEAD Morphemes To keep the MBC hierarchy variant-independent, we have also chosen a variant-independent representation of the morphemes that the MBC hierarchy maps to. We refer to these morphemes as *abstract morphemes* (AMs). The AMs are then ordered into the surface order of the corresponding concrete morphemes. The ordering of AMs is specified in a variant-independent context-free grammar. At this point, our example (1) looks like this:

(2) [Root:zhr][PAT_PV:VIII]
[VOC_PV:VIII-act] + [SUBJSUF_PV:3FS]

Note that the root, pattern, and vocalism are not ordered with respect to each other, they are simply juxtaposed. The '+' sign indicates the ordering of affixival morphemes. Only now are the AMs translated to *concrete morphemes* (CMs), which are concatenated in the specified order. Our example becomes:

(3) <zhr,V1tV2V3,iaa> +at

Simple interdigitation of root, pattern and vocalism then yields the form *iztahar+at*.

MAGEAD Rules We have two types of rules. *Morphophonemic/phonological rules* map from the morphemic representation to the phonological and orthographic representations. For MSA, we have 69 rules of this type. *Orthographic rules* rewrite only the orthographic representation. These include, for example, rules for using the gemination *shadda* (consonant doubling diacritic). For Levantine, we have 53 such rules.

For our example, we get */izdaharat/* at the phonological level. Using standard MSA diacritized orthography, our example becomes *Aizdaharat* (in transliteration). Removing the diacritics turns this into the more familiar ازدهرت *Azdhrt*. Note that in analysis mode, we hypothesize all possible diacritics (a finite number, even in combination) and

perform the analysis on the resulting multi-path automaton. We follow (Kiraz, 2000) in using a multi-tape representation. We extend the analysis of Kiraz by introducing a fifth tier. The five tiers are used as follows: Tier 1: pattern and affixational morphemes; Tier 2: root; Tier 3: vocalism; Tier 4: phonological representation; Tier 5: orthographic representation. In the generation direction, tiers 1 through 3 are always input tiers. Tier 4 is first an output tier, and subsequently an input tier. Tier 5 is always an output tier.

We implemented our multi-tape finite state automata as a layer on top of the AT&T two-tape finite state transducers (Mohri et al., 1998). We defined a specification language for the higher multi-tape level, the new MORPHTOOLS format. Specification in the MORPHTOOLS format of different types of information such as rules or context-free grammars for morpheme ordering are compiled to the appropriate LEXTOOLS format (an NLP-oriented extension of the AT&T toolkit for finite-state machines, (Sproat, 1995)). For reasons of space, we omit a further discussion of MORPHTOOLS. For details, see (Habash et al., 2005).

From MSA to Levantine and Egyptian We modified MAGEAD so that it accepts Levantine rather than MSA verbs. Our effort concentrated on the orthographic representation; to simplify our task, we used a diacritic-free orthography for Levantine developed at the Linguistic Data Consortium (Maamouri et al., 2006). Changes were done only to the representations of linguistic knowledge, not to the processing engine. We modified the MBC hierarchy, but only minor changes were needed. The AM ordering can be read off from examples in a fairly straightforward manner; the introduction of an indirect object AM, since it cliticizes to the verb in dialect, would, for example, require an extension to the ordering specification. The mapping from AMs to CMs, which is variant-specific, can be obtained easily from a linguistically trained (near-)native speaker or from a grammar handbook. Finally, the rules, which again can be variant-specific, require either a good morpho-phonological treatise for the dialect, a linguistically trained (near-)native speaker, or extensive access to an informant. In our case, the entire conversion from MSA to Levantine was performed by a native speaker linguist in about six hours. A similar but more limited effort was done to extend the Levantine system to Egyptian by introducing the Egyptian concrete morpheme for the future marker +هـ *h+* 'will'.

5. Resource Integration & Use: DIRA

DIRA (Dialectal Information Retrieval for Arabic) is a component in an information retrieval (IR) system for Arabic. It integrates the different resources created above in its pipeline. As mentioned before, one of the main problems of searching Arabic text is the diglossic nature of the Arabic speaking world. Though MSA is used in formal contexts on the Internet, e.g., in news reports, DA is dominant in user-generated data such as weblogs and web discussion forums. Furthermore, the fact that Arabic is a morphologically rich language only adds problems for IR systems. DIRA addresses both of these issues. DIRA is basically a query-term expansion module. It takes an MSA verb (and possibly some contextual material) as input and generates three types of surface forms for the search engine (the contextual material is left unchanged):

- **Mode 1:** MSA inflected forms. For example, the MSA query term أصبح *ÂSbH* 'he became' is expanded to several MSA forms including أصبحنا *ÂSbHnA* 'we became', سيصبح *sySbH* 'he will become', etc.
- **Mode 2:** MSA inflected with dialectal morphemes. It is common in DA to borrow an MSA verb and inflect it using dialectal morphology; we refer to this phenomenon as intra-word code switching. For example, the MSA query term أصبح *ÂSbH* can be expanded into هيصبح *hySbH* 'he will become' and هيصبحوا *hySbHwA* 'they will become'.
- **Mode 3:** MSA lemma translated to a dialectal lemma, and then inflected with dialectal morphemes. For example, the MSA query term أصبح *ÂSbH* can be expanded into EGY بقى *bqý* 'he became' and هيبقى *hybqý* 'he will become'.

Currently, DIRA handles EGY and LEV; with the existence of more resources for additional dialects, they will be added. The DIRA system architecture is shown in Figure 2. After submitting an MSA query to DIRA, the verb is extracted out of its context and sent to the MSA verb lemma detector, which is responsible for analyzing an MSA verb (using MAGEAD in the analysis direction) and computing its lemma (using MAGEAD in the generation direction). The next steps depend on the chosen dialects and modes. If translation to one or more dialects is required, the input lemma is translated to the dialects (Mode 3). Then, the MAGEAD analyzer is run on the lemma (MSA or DA, if translated) to determine the underlying morphemes (root and pattern), which are then used to generate all inflected forms using MAGEAD (again, which forms are generated depends on the mode). Finally, the generated forms are re-injected in the original query context (duplicates are removed).

6. Conclusions and Future Work

We presented COLABA, a large effort to create resources and processing tools for Dialectal Arabic. We briefly described the objectives of the project and the various types of resources and tools created under it. We plan to continue working on improving the resources and tools created so far and extending them to handle more dialects and more types of dialectal data. We are also considering branching into application areas other than IR that can benefit from the created resources, in particular, machine translation and language learning.

Acknowledgments

This work has been mostly funded by ACXIOM Corporation.

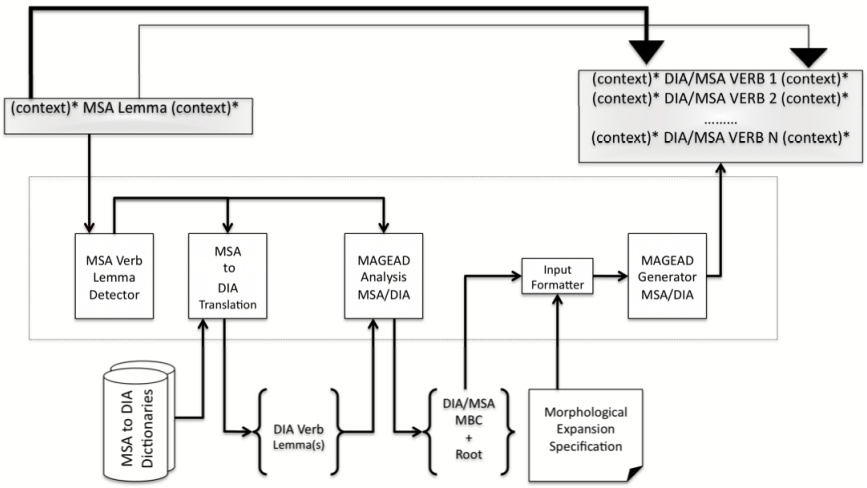

Figure 2: DIRA system architecture

7. References

Ernest T. Abdel-Massih, Zaki N. Abdel-Malek, and El-Said M. Badawi. 1979. *A Reference Grammar of Egyptian Arabic*. Georgetown University Press.

Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

Mohamed Altantawy, Nizar Habash, Owen Rambow, and Ibrahim Saleh. 2010. Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Mary Catherine Bateson. 1967. *Arabic Language Handbook*. Center for Applied Linguistics, Washington D.C., USA.

Yassine Benajiba and Mona Diab. 2010. A web application for dialectal arabic text annotation. In *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*.

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

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0.

Mark W. Cowell. 1964. *A ReferenceGrammar of Syrian Arabic*. Georgetown University Press.

Wallace Erwin. 1963. *A Short Reference Grammar of Iraqi Arabic*. Georgetown University Press.

Nizar Habash and Owen Rambow. 2006. Magead: A morphological analyzer for Arabic and its dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling-ACL'06)*, Sydney, Australia.

Nizar Habash, Owen Rambow, and Geroge Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, MI.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

N. Habash, O. Rambow, M. Diab, and R. Kanjawi-Faraj. 2008. Guidelines for Annotation of Arabic Dialectness. In *Proceedings of the LREC Workshop on HLT & NLP within the Arabic world*.

Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

H. Kilany, H. Gadalla, H. Arram, A. Yacoub, A. El-Habashi, and C. McLemore. 2002. Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22.

George Anton Kiraz. 2000. Multi-tiered nonlinear morphology using multi-tape finite automata: A case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105.

Mohamed Maamouri, Tim Buckwalter, and Christopher Cieri. 2004. Dialectal Arabic Telephone Speech Corpus: Principles, Tool design, and Transcription Conventions. In *NEMLAR International Conference on Arabic Language Resources and Tools*.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC'06*, Genoa, Italy.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 1998. A rational design for a weighted finite-state transducer library. In D. Wood and S. Yu, editors, *Automata Implementation*, Lecture Notes in Computer Science 1436, pages 144–58. Springer.

Frank Rice and Majed Sa'id. 1979. *Eastern Arabic*. Georgetown University Press.

Richard Sproat. 1995. Lextools: Tools for finite-state linguistic analysis. Technical Report 11522-951108-10TM, Bell Laboratories.

A Linguistic Search Tool for Semitic Languages

Alon Itai
Knowledge Center for Processing Hebrew
Computer Science Department
Technion, Haifa, Israel
E-mail: itai@cs.technion.ac.il

Abstract

The paper discusses searching a corpus for linguistic patterns. Semitic languages have complex morphology and ambiguous writing systems. We explore the properties of Semitic Languages that challenge linguistic search and describe how we used the Corpus Workbench (CWB) to enable linguistic searches in Hebrew corpora.

1. Introduction

As linguistics matures, so the methods it uses turn towards the empirical. It is no longer enough to introspect to gather linguistic insight. Data is required. While most search engines look for words, linguists are interested in grammatical patterns and usage of words within these patterns. For example, searching the adjective "record" followed by a noun should yield "record highs" but not "record songs"; searching the verb *to eat* (in any inflection) followed by a noun, should yield sentences such as "John ate dinner." but not "Mary ate an apple" (the verb is followed by an article, not a noun)

To answer these needs, several systems have been constructed. Such systems take a corpus and preprocess it to enable linguistic searches. We argue that general purpose search tools are not suitable for Semitic languages unless special measures are taken. We then show how to use one such tool to enable linguistic searches in Semitic Languages, with Modern Hebrew as a test case.

2. Semitic Languages

Semitic languages pose interesting challenges to linguistic search engines. The rich morphology entails that each word contains, in addition to the lemma, a large number of morphological and syntactical features: part of speech, number, gender, case. Nouns also inflect for status (absolute or construct) and possessive. Verbs inflect for person, tense, voice and accusative. Thus one may want to search for *a plural masculine noun, followed by a plural verb in past tense with accusative inflection first person singular.*

Additional problems arise by the writing system. Some prepositions and conjunctives are attached to the word as prefixes. For example, in Hebrew the word *bbit*[1] may only be analyzed as the preposition *b* (in) + the noun *bit* (house), whereas the word *bit*, which also starts with a *b* can be analyzed only as the noun *bit*, since the remainder, *it*, is not a Hebrew word. Thus to find a preposition one needs to perform a morphological analysis of the word to decide whether the first letter is a preposition or part of the lemma. Hence, in order to extract useful information the text has to be first morphologically analyzed.

All this leads to a high degree of morphological ambiguity. Ambiguity is increased since the writing systems of Arabic and Hebrew omit most of the vowels. In a running Hebrew text a word has 2.2-2.8 different analyses on the average. (The number of analyses depends on the corpus and the morphological analyzer – if the analyzer distinguishes between more analyses and if it uses a larger lexicon it will find more analyses.)

Ideally one would wish to use manually tagged corpora, i.e., corpora where the correct analysis of each word was manually chosen. However, since it is expensive to manually tag a large corpus, the size of such corpora is limited and many interesting linguistic phenomena will not be represented. Thus, one may either use automatically tagged corpora or use an ambiguous corpus and retrieve a sentence if any one of its possible analyses satisfies the query. We preferred the latter approach because of the high error rate of programs that attempt to find the right analysis in context. (An error rate of 5% per word entails that a 20 word sentence has probability $(1-0.05)^{20} \approx e^{-1}$ of being analyzed incorrectly.) Moreover, since these systems use Machine learning (HMM or SVM) (Bar Haim et al. 2005, Diab et al. 2004, Habash et al 2005), they prefer the more common structure, thus rare linguistic structures will be more likely to be incorrectly tagged. However these are exactly the phenomena a corpus linguist would like to search. Consequently, to successfully perform linguistic searches one cannot rely on the automatic morphological disambiguation and it would be better to allow all possible analyses and retrieve a sentence even if only one of the analyses satisfies the query.

[1] We use the following transliteration תשרקצפעסנמלכיטחזוהדגבא abgdhwzxTiklmnsypcqršt

3. CWB

CWB – the Corpus Workbench – is a tool created at the University of Stuttgart for searching corpora. The tool enables linguistically motivated searches, for example one may search a single word, say "`interesting`". The query language consists of Boolean combinations of regular expressions, which uses the POSIX EGREP syntax, e.g. the query

`"interest(s|(ed|ing)(ly)?)?"`

yields a search for either of the words `interest`, `interests`, `interested`, `interesting`, `interestedly`, `interestingly`.

One can also search for the lemma, say `"lemma=go"` should yield sentences containing the words `go`, `goes`, `going`, `went` and `gone`. The search can be focused on part of speech "`POS=VERB`". CWB deals with incomplete specifications by using regular expressions. For example, a verb can be subcategorized as `VBG` (present/past) and VGN (participle). The query `[pos="VB.*"]` matches both forms and may be used to math all parts of speech that start with the letters `VB`. ("." matches any single character and "*" after a pattern indicates 0 or more repetitions, thus ".*" matches any string of length 0 or more.). Finally, a query may consist of several words thus `["boy"][POS=VERB]` yields all sentences that contain the word boy followed by a verb.

To accommodate linguistic searches, the corpus needs to be tagged with the appropriate data (such as, lemma, POS). The system then loads the tagged corpus to create an index. To that end the corpus should be reformatted in a special format.

CWB has been used for a variety of languages. It also supports UTF-8, thus allowing easy processing of non Latin alphabets.

4. Creating an Index

In principle we adopted the CWB solution to partial and multiple analyses, i.e., use regular expressions for partial matches. We created composite POS consisting of the concatenation of all subfields of the analysis. For example, the complete morphological analysis of *hšxqnim* "the (male) players" is

"NOUN-masculine-plural-absolute-definite", we encode all this information as the POS of the word, [pos="*šxqnim*-NOUN-masculine-plural-absolute-definite"], the lemma is šxqnim , the main POS is noun, the gender masculine, the number plural, the status absolute and the prefix h indicates that the word is definite. We included the lemma, since each analysis might have a different lemma.

To accommodate for multiple analyses, we concatenate all the analyses (separated by ":"). For example,

mxiibim
:*mxaiib*-ADJECTIVE-masculine-plural-abs-indef
:*xiib*-PARTICIPLE-Pi'el-*xwb*-unspecified-masculine-plural-abs-indef
:*xiib*-VERB-Pi'el-*xwb*-unspecified-masculine-plural-present:PREFIX-*m*-preposition
:*xiib*-NOUN-masculine-plural-abs-indefinite
:PREFIX-*m*-preposition- *xiib*-ADJECTIVE- masculine-plural -abs-indef:

The analyses are:

1. The adjective *mxiib*, gender masculine, number plural, status absolute and the word is indefinite;
2. The verb *xiib* it is a participle of a verb whose binyan (inflection pattern of verb) is Pi'el, the root is *xwb*, the person is unspecified, gender masculine, number plural, the type of participle is noun, the status absolute and the word is indefinite.
3. A verb whose root is *xwb*, binyan Pi'el, person unspecified, number plural and tense present.
4. The noun *xiib*, prefixed by the preposition *m*.
5. The adjective *xiib*, prefixed by the preposition *m*.

Thus one can retrieve the word by any one of the queries by POS:
[POS=".*-ADJECTIVE-.*"], [POS=".*-PARTICIPLE-.*"], [POS=".*-VERB-.*"], [POS=".*-NOUN-.*"].
However, one may also specify additional properties by using a pattern that matches subfields:
[POS=".*PREFIX-[^:]*preposition[^:]*-NOUN-.*"]
indicating that we are searching for a noun that is prefixed by a preposition. The sequence [^:]* denotes any sequence of 0 or more characters that does not contain ":" and is used to skip over unspecified sub-fields. Since the different analyses of a word are separated by ":" and ":" cannot appear within an analysis, the query cannot be satisfied by matching the part of the query by one analysis and the remainder of the query by a subsequent analysis.

To create an index from a corpus, we first run the morphological analyzer of MILA (Itai and Wintner 2008) that creates XML files containing all the morphological analyses for each word. We developed a program to transform the XML files to the above format, which conforms to CWB's index format. Thus we were able to create CWB files.

Our architecture enables some Boolean combinations. Suppose we wanted to search for a two-word expression noun-adjective that agree in number. We therefore could require that the first word be a singular noun and the second word a singular adjective or the first word is a plural noun and the second word a plural adjective. The query

```
([pos=".*NOUN-singular-.*"]
[pos=".*ADJECTIVE-singular-.*"])
|([pos=".*NOUN-plural-.*"]
 [pos=".*ADJECTIVE-plural-.*"])
```

However, one must be careful to avoid queries of the type `[pos=".*NOUN.*" & pos=".*-singular-.*"]` since then we might return a word that has one analysis as a plural noun and another analysis as a singular verb.

5. Performance

To test the performance of the system we uploaded a file of 814,147 words, with a total of 1,564,324 analyses, i.e., 2.36 analyses per word. Table 1 shows a sample of queries and their performance. The more general the queries the more time they required. However, the running time for these queries is reasonable. If the running time is linear in the size of the corpus, CWB should be able to support queries to 100 million word corpora.

One problem we encountered is that of space. The index of the 814,147 word file required 25.2 MB. Thus each word requires about 31 bytes. Thus a 100 Million word corpus would require a 3.09 Gigabyte index file.

6. Writing Queries

Even though it is possible to write queries in the above format we feel that it is unwieldy. First the format is complicated and one may easily err. However more importantly, in order to write a query one must be familiar with all the features of each POS and in which order they appear in the index. This is extremely user-unfriendly and we don't believe many people will be able to use such a system.

To overcome this problem, we are in the process of creating a GUI which will show for each POS the appropriate subfields and once a subfield is chosen a menu will show all possible values of that subfield. Unspecified subfields will be filled by placeholders. The graphic query will then be translated to a CWB query and the results of this query will be presented to the user. We believe that the GUI will also be helpful for queries in languages that now use CWB format.

Regular Expression	Time (sec)	Output File (KB)
[pos=".*-MODAL-.*"] [pos=":היה-.*"][pos=".*-VERB-[^:]*-infinitive:.*"];	0.117	13
[word="על"][word="מנת"][pos=".*-VERB-[^:]*-infinitive:.*"];	0.038	28
[word="על"][word="מנת"][pos=".*PREFIX-ש.*"];	0.025	5
[pos=".*:הלך-[^:]*-present:.*"] [pos=".*-VERB-[^:]*-infinitive:.*"];	0.099	2
[word="בית"][pos=":ספר.*"];	0.017	7
[word="בית"][pos=".*:ספר[^:]*-SUFFIX-possessive-.*"];	0.014	1
"כותב";	0.009	
[pos=".*:[^:]*-VERB-[^:]*:.*"];	0.569	
".*";	1.854	
[pos=".*"];	1.85	
".*"; [pos=".*"];	3.677	
All the previous regular expressions concatenated	7.961	
("כותב" | [pos=".*:[^:]*-VERB-[^:]*:.*"] | ".*" | [pos=".*"]);	2.061	
([pos=".*:[^:]*-VERB-[^:]*:.*" & word="כותב" & word=".*" & pos=".*"]);	0.168	

Table 1: Example queries and their performance.

7. Conclusion

Until now Linguistic searches were oriented to Western languages. Semitic languages exhibit more complex patterns, which at first sight might require designing entirely new tools. We have showed how to reuse existing tool to efficiently conduct sophisticated searches.

The interface of current systems is UNIX based. This might be acceptable when the linguistic features are simple, however, for complex features, it is virtually impossible to memorize all the possibilities and render the queries properly. Thus a special GUI is necessary.

8. Acknowledgements

It is a pleasure to thank Ulrich Heid, Serge Heiden and Andrew Hardie who helped us use CWB. Last and foremost I wish to thank Gassan Tabajah whose technical assistance was invaluable.

9. References

Official Web page of CWB: http://cwb.sourceforge.net/

Bar Haim, R., Sima'an, K. and Winter, Y. (2005). Choosing an Optimal Architecture for Segmentation and POS-Tagging of Modern Hebrew. *ACL Workshop on Computational Approaches to Semitic Languages.*

Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. *Linguistic Data Consortium catalog number LDC2002L49, ISBN 1-58563-257-0.*

Buckwalter, T. (2004). Buckwalter Arabic Morphological Analyzer Version 2.0. *Linguistic Data Consortium catalog number LDC2004L02, ISBN 1-58563-324-0.*

Christ, O. (1994). A modular and flexible architecture for an integrated corpus query system. In *Papers in Computational Lexicography (COMPLEX '94),* pp. 22--32, Budapest, Hungary.

Christ, O. and Schulze, B. M. (1996). Ein flexibles und modulares Anfragesystem für Textcorpora. In H. Feldweg and E. W. Hinrichs (eds.), *Lexikon und Text*, pp. 121--133. Max Niemeyer Verlag, Tübingen.

Diab, M., Hacioglu, K. and Jurafsky, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *HLT-NAACL: Short Papers*, pp. 149--152.

Habash, N. and Rambow, O. (2005). 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*, pp. 573--580, Ann Arbor.

Hajič, J. (2000). Morphological tagging: data vs. dictionaries. In *Proceedings of NAACL-ANLP,* pp. 94--101, Seattle, Washington.

Hajič, J. and Barbora Hladká, B. (1998). Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of COLING-ACL* 1998. pp. 483--490, Montreal, Canada

Itai, A. and Wintner, S. (2008). Language Resources for Hebrew. *Language Resources and Evaluation,* 42, pp. 75--98.

Lee, Y-S. et al. (2003). Language model based Arabic word segmentation. In *ACL* 2003, pp. 399--406.

Segal, E. (2001). Hebrew morphological analyzer for Hebrew undotted texts. M.Sc. thesis, Computer Science Department, Technion, Haifa, Israel.

Algerian Arabic Speech Database (ALGASD): Description and Research Applications

G. Droua-Hamdani 1, S. A. Selouani 2, M.Boudraa 3

1 Speech Processing Laboratory (TAP), CRSTDLA, Algiers, Algeria.

2 LARIHS Laboratory. University of Moncton, Canada.

3 Speech Communication Laboratory, USTHB, Algiers, Algeria.

gh.droua@post.com, selouani@umcs.ca, mk.boudraa@yahoo.fr

Abstract

This paper presents Algerian Speech Database (ALGASD) for standard Arabic language and relative research applications. The project concerns 300 Algerian native speakers whom are selected statistically from 11 regions of the country. These different areas are assumed representing the principal variations of pronunciations denoted between the populations.

ALGASD took into consideration many features as: gender, age and education level of every speaker. The basic text used to elaborate the database is constituted by 200 phonetically balanced sentences. Number of recordings achieves to 1080 read sentences.

ALGASD provides an interesting set of perspectives for speech science applications, such as: automatic speech recognition, acoustic phonetic analysis, prosodic researches; etc. This voice bank is used until now in several studies like rhythm analysis of Algerian speakers. It is also used in training and testing phases of speech recognition systems, etc.

1. Introduction

The utilization of databases in the development of "human-machine" applications is very important as for: Text-To-Speech systems, Speech Recognition systems, etc. These last decades, many different databases were realized. They can be:

- Multilingual which constitute important projects containing several languages, (Van den Heuvel & & Galounov & Tropf , 1998; Schultz & Waibel, 1998; Roach & Vicsi, 1996; Chan & al., 1995; Siemund & al., 2000) or monolingual limiting themselves to only one language (Timit, 1990; Vonwiller & al., 1995; ELRA Ref120);
- Official / dialectal languages (Gopalakrishna & al., 2005);
- Reserved for a restricted domain or not such as: telephony (Petrovska & al., 1998; Zherng & al., 2002) etc.
- Dealing with continuous speech, read texts, etc (Siemund & al., 2000; ELRA Ref120; Gopalakrishna & al., 2005).

In comparison with the multitude of oral corpora realized for European and Asians languages, the corpora dedicated to the Arabic language and to its dialects are less frequent. It exists to our knowledge: the corpus of LDC for the spontaneous phone word realized by Egyptian, Syrian, Palestinian and Jordanian speakers (LDC), ELRA's corpora for the Standard Arabic read by Moroccan speakers [LDC], the oral corpus of GlobalPhone (Siemund & al., 2000), Nemlar corpus recorded from radio stations (Choukri & Hamid & Paulsson, 2005) and finally SAAVB for the Saudi accent (Mohamed &. Alghamdi & Muzaffar, 2007).

2. Standard Arabic in Algeria's Languages

ALGASD project consists on conception and realization of Algerian voice bank with the Standard Arabic as the substratum.

Situated in the north of Africa, Algeria extends over the vast territory of 2 380 000 km^2 occupied by about 34.8 million inhabitants. The majority of them are concentrated in the north.

Algerians' official language is Standard Arabic (SA). It is used for all administrative tasks (government, media, etc). It is taught approximately for 13 years to children from 6 to17 years old in three academic levels: primary, middle and secondary school. However, written SA differs very substantially from Algerian spoken languages (mother tongues). Indeed, approximately 72 % of the population speaks in their daily life the *Darija*, Algerian Arabic dialects and 28% of them have a second mother tongue called *Tamazight* which is Berber language.

Algerian Dialects are variants of SA stemming from the ethnic, geographical and colonial occupiers influences as Spanish, French, Turkish, Italian, etc. While, Algiers *Darija* is influenced by both Berber and Turkish, Constantine dialect is affected by Italian, Oran by Spanish, Tlemcen by Andalusia Arabic, etc. As a result, within Algerian Arabic itself, there are significant local variations (in pronunciation, grammar, etc.) observed from town to town even they are near to each other (Taleb, 1997; Marçais, 1954; Caubet, 2001).

These two native languages (*Darija* and *Tamazight*) constitute so the principal oral communication between Algerians. In addition, the third language used by some Algerians is French language though it has no official status, but it is still widely used by government, culture,

media (newspapers), universities, etc. (Arezki, 2008; Cheriguen,1997).

3. Corpus Design

Text material of ALGASD is built from 200 Arabic Phonetically Balanced Sentences (APBS) (Boudraa, & Boudraa & Guerin 1998). From which we conceived three types of corpora. Every corpus aims to provide us a specific acoustic-phonetic knowledge. *Common Corpus (Cc):* is used to list a maximum of dialectal variations of pronunciation observed among Algerians. It is composed of two utterances of APBS read by all speakers. *Reserved Corpus (Cr):* brought all existing phonetic oppositions in the Arabic language. It is endowed with 30 sentences of APBS which are divided into 10 texts of 3 sentences and sheared between groups of speakers. In order to increase some consonants' occurrences, we broke some times the balance. *Individual Corpus (Ci):* is constituted of 168 remaining sentences. They are used to gather maximum of contextual allophones.

To elaborate ALGASD, we selected 300 Algerian speakers from 11 regions of the country which mapped the most important variation of pronunciations between inhabitants. All participants are native speakers and had grown up in or near localities selected for this research. According to the most recent census of inhabitants available in ONS web site (ONS)**,** we distributed statically all speakers between these areas with regard to the real number and gender of inhabitants for each region (Table.1).

	Regions	**Female**	**Male**	**T. Speaker/ Region**
R1	Algiers	40 (50%)	40 (50%)	80 (27%)
R2	Tizi Ouzou	17 (50%)	17 (50%)	34 (11%)
R3	Medea	13 (52%)	12 (48%)	25 (8%)
R4	Constantine	13 (52%)	12 (48%)	25 (8%)
R5	Jijel	09 (50%)	09 (50%)	18 (6%)
R6	Annaba	09 (52%)	08 (48%)	17 (6%)
R7	Oran	19 (50%)	19 (50%)	38 (13%)
R8	Tlemcen	13 (50%)	13 (50%)	26 (9%)
R9	Bechar	04 (52%)	03 (48%)	07 (2%)
R10	El Oued	08 (50%)	08 (50%)	16 (5%)
R11	Ghardaïa	07 (50%)	07 (50%)	14 (5%)
Total	11	152 (51%)	148 (49%)	300 (100%)

Table 1: Speakers' distribution in ALGASD

4. ALGASD Features

The speaker profile used in database takes into consideration age and education level of every speaker. We suggested, so, for these two features respectively three different categories: (18-30/ 30-45/ +45) and (Middle/Graduate/Post Graduate).

Recordings are made in quiet environments well known by the speakers. The same conditions of sound recording are respected for all regions. We selected the best reading and deleted all sentences which contained hesitations, re-recorded utterances which were not spoken clearly, correctly, too soft or too loud. The average duration of sentences is about 2.8 seconds. Rate of recording is normal. The sound files are in wave format, coded on 16 bits and sampling at 16 KHz.

5. Recordings

Recordings were preceded as follow:

Every 3 texts of Cr are distributed periodically on the 11 regions. In the beginning, we shared these 3 texts and gave them to 3 speakers (2 male /1 female), excepted for R9, where it was endowed only by 2 texts for 2 speakers (1 male/1 female). But after, we augmented the number of recordings by increasing the number of speakers for each region (Table.2). Total speakers and recordings reached then respectively to 86 and 258 sound files.

Cc text material was read by all speakers of ALGASD (300 speakers). Number of readings achieved to 600 recordings. As regards to *Ci*' text, we realized 2 different sub-sets of recordings: the first one contains 32 utterances read by all speakers of Cr corpus. The second one is constituted of 136 sentences statistically distributed between 136 other speakers for all regions. From this operation, two sentences were remaining. We added them to R9 texts because it contained the less number of speakers

	M	**F**	**Recordings**
R1	12	11	69
R2	5	5	30
R3	4	3	21
R4	4	3	21
R5	3	2	15
R6	3	2	15
R7	6	5	33
R8	4	3	21
R9	1	1	6
R10	3	2	15
R11	2	2	12
11	47	39	258
	86 speakers		

Table 2: Recordings of Cr corpus

In conclusion, 28 % of speakers read 6 sentences, 45 % read 3 sentences and 26% read only 2 ones. Total number of ALGASD recordings reached to 1080 (Table 3).

Corpora	N° utterances	speakers	Total
Cc	2	300	600 (55.5%)
Cr	30	86	258 (24.0%)
Ci	168	222	222 (20.5%)
TOTAL	**200**	**300**	**1080** (100%)

Table 3: Total corpora and speakers of ALGASD

6. Research Applications of ALGASD speech corpus

ALGASD corpus is characterized by many aspects as: a high quality of recordings, a large number of speakers, speaker's features which reflect many differences due to region, age, gender, education levels and the dialect varieties. All these characteristics provide an interesting set of perspectives for speech science applications, such as: automatic speech recognition, acoustic phonetic analysis, perceptual experiments to study classification of the different regional varieties spoken within Algeria SA, prosodic studies as rhythm, comparison of Algerian SA with Arabic of Maghreb countries or eastern ones, etc.

ALGASD database was used until now in many studies as: statistical study of qualitative and quantitative vocalic variations according to education levels of Algiers speakers (Droua-Hamadani, & Selouani & Boudraa & Boudraa , 2009); Location of Algerian Standard Arabic Rhythm between stressed languages (to appear); Impact of education levels on duration and rhythm of Algerian modern Standard Arabic (to appear). By respecting some recommendations in the selection and the distribution of both sound material and speakers, we built from ALGASD two required corpora to train and test speech recognition system for Algerian Standard Arabic (Droua-Hamadani, & Selouani & Boudraa & Boudraa , 2009).

7. References

Arezki, A. (2008). Le rôle et la place du français dans le système éducatif algérien. Revue du Réseau des Observatoires du Français Contemporain en Afrique, N° 23. pp 21-31.

Boudraa, M. & B. Boudraa, B. & Guerin, B. (1998). Twenty Lists of Ten Arabic Sentences for Assessment. ACUSTICA Acta-acustica. Vol.84.

Caubet, D. (2001). Questionnaire de diactologie du Maghreb (D'après les travaux de W Marçais, M Cohen, G.S Colin, J Cantineau, D. Cohen, Ph. Marçais. S Levy, ect.). Estudios de dialectologiia norteafricana y andalusi, pp. 73-92.

Chan, D. & al. (1995) EUROM- a Spoken Language Resource for the EU. Eurospeech 9. Proceedings of the 4th European Conference on Speech Communication and Technology. Madrid, Spain.

Cheriguen, F. (1997). Politique linguistique en Algérie, in Mots, Les langages du politique, n52, pp. 62-74.

Choukri, K. Hamid, S. Paulsson, N. (2005). Specifiaction of the Arabic Broadcast News Speech Corpus NEMLAR: http://www.nemlar.org.

Droua-Hamadani, G. & al. (2009). ALGASD PROJECT: Statistical Study of Vocalic Variations according to Education Levels of Algiers Speakers., Intonational Variation in Arabic Conference IVA09, York, (England).

Droua-Hamadani, G. & Selouani, S.A. & Boudraa, M. (2009). ALGASD Algerian voice bank project ALGASD's adaptation for continuous speech recognition system. The 5th International Conference on Computer Science Practice in Arabic (CSPA '09 AICCSA09- IEEE), Rabat (Marroco).

Gopalakrishna, A. & al (2005). Development of Indian Language Speech Databases of Large Vocabulary Speech Recognition Systems. Proceedings of International Conference On Speech an Computer (SPECOM), Patras, Greece.

Linguistic Data Consortium (LDC): http://www.ldc.upenn.edu.

Marçais, P. Textes arabes de Djidjelli (1954). Presse universitaire de France.

Mohamed, A. M. Alghamdi, M. & Z. Muzaffar, Z. (2007). Speaker Verification Based on Saudi Accented Arabic Database. International Symposium on Signal Processing and its Applications in conjunction with the International Conference on Information Sciences, Signal Processing and its Applications. Sharjah, United Arab United Arab Emirates.

National Office of Statistics (ONS): http://www.ONS.dz.

Petrovska, D. & al (1998). POLYCOST: A Telephone-Speech Database for Speaker recognition. Proceedings RLA2C ("Speaker Recognition and its Commercial and Forensic Applications"), Avignon, France, pp. 211-214. (http://circhp.epfl.ch/polycost).

REF120 corpus available in European Language Resources Association: http://www.icp.grenet.fr/ELRA.

Roach, P. & Vicsi, K. (1996). BABEL An Eastern European Multi-Language Database. COST249 meeting Zurich.

Schultz, T. & Waibel, A.GlobalPhone (1998). Das Projekt GlobalPhone: Multilinguale Spracherkennung

Computers, Linguistics, and Phonetics between Language and Speech, Bernhard Schröder et al (Ed.) Springer, Berlin 1998, ISBN Proceedings of the 4th Conference on NLP - Konvens-98, Bonn, Germany.

Siemund, R. & al. (2000). SPEECON – Speech Data for Consumer Devices. Proceedings of LREC 2000.

Taleb Ibrahimi, K. (1997) Les Algériens Et Leur(s) Langue(S), Eléments pour une approche sociolinguistique de la société algérienne. Les editions EL HIKMA. Deuxième Edition.

Texas Instrument and Massachusetts Institute of Technology corpus (TIMIT) (1990): Acoustic-Phonetic Continuous Speech Corpus. DMI.

Tseng, C. & Lee, W. & Huang, F. (2003). Collecting Mandarin Speech Databases for Prosody Investigations. The Oriental COCOSDA. Singapore.

Van den Heuvel, H. & Galounov, V. & Tropf, H.S (1998). The SPEECHDAT (E) project: Creating speech databases for eastern European languages. Proceedings Workshop on Speech Database Development for Central and Eastern European Languages. Granada, Spain.

Vonwiller, J. & al (1995). Speaker and Material Selection for the Australian National Database of Spoken Language. Journal of Quantitative Linguistics, 2: 177-211.

Zherng, T.F & al (2002). Collection of a Chinese Spontaneous Telephone Speech Corpus and Proposal of Robust Rules for Robust Natural Language Parsing. Joint International Conference of SNLP- O-COCOSDA, Hua Hin, Thailand.

Integrating Annotated Spoken Maltese Data into Corpora of Written Maltese

Alexandra Vella†*, Flavia Chetcuti†, Sarah Grech†, Michael Spagnol‡
University of Malta†, University of Cologne*, University of Konstanz‡
alexandra.vella@um.edu.mt, fchetcuti@hotmail.com, sgrec01@um.edu.mt, michael.spagnol@uni-konstanz.de

Abstract

Spoken data features to a lesser extent in corpora available for languages than do written data. This paper addresses this issue by presenting work carried out to date on the development of a corpus of spoken Maltese. It outlines the standards for the PRAAT annotation of Maltese data at the orthographic level, and reports on preliminary work on the annotation of Maltese prosody and development of *ToBI*-style standards for Maltese. Procedures being developed for exporting PRAAT TextGrid information for the purposes of incorporation into a predominantly written corpus of Maltese are then discussed. The paper also demonstrates how characteristics of speech notoriously difficult to deal with have been tackled and how the exported output from the PRAAT annotations can be enhanced through the representation also of phenomena, sometimes referred to as "normal disfluencies", which include "filled pauses" and other vocalisations of a quasi-lexical nature having various functions of a discourse-management type such as "backchannelling".

1. Introduction

Annotation of spoken, as compared to written data, tends to feature to a lesser extent in corpora available for languages. For instance, the British National Corpus, "a 100 million word collection of spoken and written language from a wide range of sources, designed to represent a wide cross-section of British English" (British National Corpus), contains only about 10% of spoken data. One reason for the lesser inclusion of spoken data into corpora is the greater degree of pre-processing work which needs to be done to it before it can be included, as text, into a corpus (Gibbon et al., 1997).

The purpose of this contribution is threefold. Firstly it reports on the current state and continuing development of the corpus of spoken Maltese and its annotation, as well as on the development of standards and guidelines for this (Vella et al., 2008). Availability of such standards and guidelines should enable the training of annotators and allow for more spoken data to be prepared for inclusion in corpora of Maltese. Second, it demonstrates what procedures need to be developed in order for the PRAAT (Boersma & Weenick) TextGrid annotations available to date to be converted into "running text". Such conversion is important since it will allow incorporation of the annotations carried out into other corpora of Maltese developed in the context of projects such as *MaltiLex* and *MLRS* (Dalli, 2001; Rosner et al., 1998; Rosner et al., 2000; Rosner, 2009). Third, it outlines continuing linguistic analysis of features of (quasi-)spontaneous speech known to be particularly difficult to deal with. The features in question include intonation, but also phenomena of the sort sometimes referred to as "normal disfluencies" which include, amongst others, "repetitions", "repairs" and "filled pauses" (Cruttenden, 1997), as well as other features which have been shown to serve various functions of a discourse-management type such as "backchannelling" (original use due to Yvnge, 1970). Work on analysis of these features is ongoing (Vella et al., 2009).

2. Work to date

Work on the annotation of spoken Maltese carried out within the context of two projects, *MalToBI* and *SPAN*, has produced preliminary guidelines for the annotation of spoken data from Maltese together with a small amount of annotated data. Annotation is being done using PRAAT and has to date concentrated on quasi-spontaneous Map Task dialogue data from the *MalToBI* corpus (Vella & Farrugia, 2006). The available annotations have a structure involving different types of information included in separate TIERS. The tiers in the current annotation are the following:

1. SP(eaker)1 and SP(eaker)2;
2. Br(eak)-Pa(use)-O(verlap)s;
3. T(arget)I(tem)s;
4. F(illed)P(ause)s;
5. MISC(ellaneous).

Three further tiers are also included in the TextGrids for data for which prosodic annotation has been carried out. These are as follows:

6. Tone;
7. Prom(inence);
8. Functions.

Standards for prosodic annotations are still undergoing development. Standards and guidelines for orthographic annotation of spoken data, by contrast, are at an advanced stage of development. A detailed description of the standards and guidelines which have been used in producing the annotations available to date is given in Section 3 below. Procedures which have started being developed to allow incorporation of the annotations of the spoken data into corpora consisting mainly of written Maltese are then discussed in Section 4. Lastly Section 5 describes and discusses how we annotated features of speech which are particularly difficult to handle in that they have no direct and/or obvious "written" correlate. Such features include prosody (which will only be discussed briefly here), but also phenomena sometimes referred to as "normal disfluencies" (see Cruttenden, 1997 and above). Particular attention will be given to vocalisations sometimes classified as "filled pauses", as

well as to "backchannels" or features involved in providing feedback to other participants in a dialogue. Some concluding remarks are provided in Section 6.

3. Structure of the annotations

As mentioned in Section 2, information of different types is included in separate tiers in the PRAAT annotations carried out. A sample of a very short extract from one recording, together with the associated annotation, is shown in Figure 1 below.

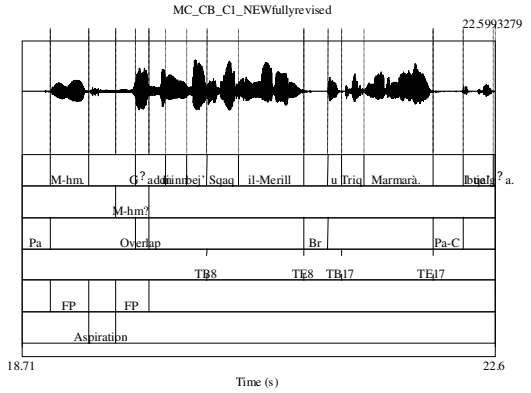

Figure 1: Sample excerpt from MC_CB_C1

3.1 Annotation tiers

The standards used in carrying out the annotations are summarised below in Subsections 3.1.1 – 3.1.4. Subsections 3.1.1 – 3.1.3 deal with the **SP1** and **SP2**, **Br-Pa-Os** and **FPs** tiers respectively, while Subsection 3.1.4 deals with the remaining **TIs** and **MISC** tiers, as well as, briefly, with the prosodic **Tone**, **Prom** and **Functions** tiers.

3.1.1. SP1 and SP2 tiers

The word-by-word annotation makes use of standard orthography, including the new spelling rules published by *Il-Kunsill Nazzjonali tal-Ilsien Malti* in 2008 and all Maltese characters use Unicode codification (see Akkademja tal-Malti, 2004; Kunsill tal-Malti, 2008). In a number of cases, however, there is some variation with respect to regular standard orthography, as it is considered important for the word-by-word annotation to provide as close a record as possible to what was actually said. Thus, for example, in cases of elision of different sorts, a convention similar to that used in standard orthography (e.g. *tazz'ilma* for *tazza ilma* 'a glass of water'), that is the use of an apostrophe, is extended to include initial elision, e.g. *'iġifieri* for *jiġifieri* 'that is to say'.[1] There are also instances of insertions. In such cases, inserted segments are added to the transcription in square brackets (e.g. *nagħmlu [i]l-proġett* 'we [will] perform the task').

Capitalisation follows punctuation rules in Maltese. The first letter of target items (on which, see also Subsection 3.1.4) is always capitalised. When lexical stress in target items is misplaced, the syllable which has been stressed is capitalised in the annotation, e.g. the expected position of stress in the proper noun ***PER****gola* in the target item *Hotel Pergola* 'Hotel Pergola' is antepenultimate; capitalisation in the TextGrid annotation *Per**GO**la* indicates that stress was assigned penultimately by the speaker in this instance.[2]

Sentential punctuation marks such as question marks (?) and full-stops (.) are included in the annotation, and are generally used in line with punctuation conventions rather than to indicate a fall or rise in pitch. Final punctuation marks such as exclamation marks (!), ellipsis (…), quotation marks (' ', " "), etc., by contrast, have not been included in the annotation. The punctuation marks in this group are intended to indicate, in written text, the presence of elements typical of speech. Specifically, the exclamation mark indicates use of intonation of a particularly "marked" kind, whilst ellipsis often indicates a pause in speech or an unfinished sentence. Both these elements are catered for in tiers other than the SP1 and SP2 tiers (see Subsection 3.1.4 and 3.1.2 respectively). Quotation marks in written texts often indicate direct, as opposed to indirect speech or narrative, not a relevant factor with respect to the annotation standards being discussed given that the texts in question consist solely of speech. Hyphens (-), accents (`) and apostrophes (') are used in the normal way as for written Maltese. Note that apostrophes are also used to indicate elision, as noted above. Internal punctuation marks such as dashes (–), semi-colons (;), colons (:), and particularly commas (,), an important element of punctuation in written texts, are avoided although also catered for in the annotations (see Subsection 3.1.2). Such punctuation marks sometimes coincide with the location of phrase boundaries of different sorts, but do not always do so. Their use is not as clearly regulated as is that of other punctuation marks, and therefore would give poor results in terms of inter-transcriber reliability.

Phrasal units involving a determiner and a noun or adjective (e.g. *il-bajja* 'the bay', *il-kbir* 'big, lit. the big', etc.), as well as units with a particle plus determiner and noun or adjective (*fid-direzzjoni* 'in the direction (of)', *tar-re* 'of the king', etc.) are segmented together. Simple particles, on the contrary, are segmented as separate expressions from the word they precede (e.g. *ta' Mejju* 'of May', *fi Triq Ermola* 'in Ermola Street', etc.). Additional conventions used which are at odds with standard punctuation are question marks at the beginning of a word to mark a dubious or unclear expression (e.g. *?Iwa/Imma* 'yes/but'), asterisks immediately before a word to mark an ungrammatical or non-existent expression in the language (e.g. *il-bajja *tar-Ray* lit. 'Ray's the bay') as

[1] Where possible, examples provided are taken from the Map Task annotations carried out.

[2] Where an indication of lexical stress is necessary in the above, the syllable in question is shown in bold capitals.

used in linguistics to indicate unacceptability, and slashes on both sides of a word to indicate non-Maltese words (e.g. */anticlockwise/*).

3.1.2. Br-Pa-Os tier

The Br-Pa-Os tier is used to indicate the presence of breaks and pauses, as well as that of overlap, in the dialogues. Examples of the distinctions made, together with a description of specific characteristics in each case, are given in Table 1 below.

Examples	Characteristics
Triq Mar<...> **Br** *Triq Mannarino.* 'Mar<...> Street **Br** Mannarino Street.'	False start or repair truncation and correction unexpected
Għaddi minn bej' Sqaq il-Merill **Br** *u Triq Marmarà.* 'Go between Merill Alley **Br** and Marmarà Street.'	Intra-turn break within constituent unexpected
Għaddi **Br** *minn bejniethom.* 'Go **Br** between them.'	Intra-turn break before adverbial less unexpected
Mela **Br** *tibda mill-Bajja ta' Ray.* 'So **Br** begin at Ray's Bay.'	Similar to comma break before main clause expected
u Triq Marmarà. **Pa** *Ibqa' tielgħa.* 'and Marmarà Street. **Pa** Keep walking upwards.'	Intra-speaker; full-stop break across sentences expected
SP1: *Ibqa' tielgħa*. **Pa-C** SP2: *Sewwa*. 'Keep walking upwards. **Pa-C** Good.'	Inter-speaker; full-stop break across sentences expected
SP2: *M-hm?* SP1: *Għaddi minn bejn Sqaq il-...* 'M-hm? **O** Go between Merill Alley...'	Inter-speaker overlap –

Table 1: Examples of Br-Pa-O distinctions made

Differentiation between breaks and pauses is based on a broad distinction between intra-sentence gaps, labelled **Break**, and inter-sentence ones, labelled **Pause**. Transcribers were instructed to allow intonation, as well as their intuitions, to inform their decisions. The distinction between **Break** and **Pause**, correlates, roughly speaking, with the comma vs. full-stop distinction made in writing. Unexpected intra-speaker mid-turn pauses associated with "normal disfluency"-type phenomena mentioned earlier (see Sections 1 and 2) are however also labelled as **Break**. Within speaker pauses across sentences are distinguished from those across speakers by means of the label **Pause** vs. **Pause-C**(hange). A study of both the distribution and durational characteristics of breaks vs. pauses is planned. Such a study is expected to throw light on the nature of different types of phonological boundaries and related boundary strength in Maltese (but see also Section 3.1.4 below).

3.1.3. The FPs tier

This tier is a very important part of the annotations. It is used to note the position of any "non-standard forms" transcribed in the **SP1** and **SP2** tiers, such forms being roughly defined as "forms not usually found in a dictionary". The **FPs** tier is extremely useful to the phonetician using PRAAT as her/his main analysis tool since it increases searchability (but see also Subsection 3.1.4).

One of the difficulties encountered in the annotation of "non-standard forms" is that such forms have no clearly recognisable "standard" representation, something which can prove problematic even to established writers . In fact, of the forms whose occurrence is noted in the FP tier, only six, namely "***e***", "***eħe***", "***eqq***", "***ew***", "***ħeqq***" and "***ta***" are listed in Aquilina's (1987/1990) dictionary. In addition, other researchers refer to different forms or to similar forms having functions that seem to be different to the ones in the data analysed (see, e.g., Borg & Azzopardi-Alexander, 1997; Mifsud & Borg, 1997).

Forms of the sort whose occurrence is noted in this tier are typically found in spontaneous speech and include phenomena such as "repetitions", "repairs" and "filled pauses" (mentioned earlier and reported in Cruttenden, 1997). Such phenomena often serve clear functions and have their own specific characteristics, phonetic, including prosodic, as well as otherwise (see, e.g., Shriberg, 1999).

As things stand at present in fact, the **FPs** tier conflates into one relatively undifferentiated group, a number of different phenomena.[3] Preliminary analysis of elements included in this tier reported by Vella et al. (2009) makes possible a distinction between "real" *filled pauses* (FPs) and other phenomena – the latter will also be discussed in this Subsection.

Although consensus amongst researchers on what exactly "counts" as an FP is limited, linguists usually agree that FPs are discourse elements which, rather than contributing information, "fill" silences resulting from pauses of various sorts. They also agree that such elements can contribute meaning and/or communicative function but do not always do so, and that they often have a role to play in the organisation of discourse.

Preliminary analysis of the forms flagged in the **FPs** tier in the data annotated included a durational, distributional and phonetic (particularly prosodic) study of the forms. One outcome of the durational study is that it has made possible the standardisation of annotation guidelines for the various "non-standard forms" found in the data. To give one example, the original annotations include three "different" forms: *e*, *ee* and *eee*. The durational study carried out however suggests that the different labels do not in fact correlate with a difference in the duration of the entities in question: instances transcribed as *eee* are not in fact longer than instances transcribed as *ee*, which are not,

[3] It is possible that the FP tier will in fact be renamed in subsequent annotation work.

in turn, longer than instances transcribed as *e*: the one label *eee* is therefore being suggested as the "standard" for all occurrences of this type of FP and the original annotations have been amended accordingly. References below are to the labels as amended rather than to the original labels.

The analysis mentioned above has led to the identification of a number of "real" FPs in Maltese, similar to FPs described for other languages. These are *eee*, *mmm* and *emm*, all of which contain the element/s [e] and [m]. The analysis also noted two other "forms" of FPs annotated in the data, namely *ehh* and *ehm*. While the latter may be phonetic variants of the above-mentioned *eee* and *emm*, instantiations of *ehh* and *ehm* in the data annotated are significantly longer than their *eee* and *emm* counterparts. They also appear to have an element of "glottalisation" not normally characteristic of instances of *eee* and *emm*.[4]

The distributional analysis of the "real" FPs *eee*, *mmm* and *emm* suggests that, overall, there is a very high tendency for silence to occur, to the left, to the right, or on both sides of these FPs. A slightly greater tendency for these kind of FPs (particularly *eee* and *mmm*) to occur following a silence, rather than preceding one is exhibited. Analysis of the intonation of the "real" FPs identified is still ongoing, however, a preliminary characterisation of the intonation of such forms is one involving a long period of level pitch around the middle of the speaker's pitch range (see also Vella et al., 2009).

A number of phenomena other than "real" FPs also occur in the data. The most important of these is a highly frequent class of forms involving "quasi-lexical" vocalisations such as *m-hm* and *eħe/aħa/ija/iwa*, which tend to have clear meanings (perhaps similar to Cruttenden's 1997:175 "intonational idioms"). The form *m-hm* is particularly worthy of note. This was originally transcribed as *mhm* in the data annotated. The main reason for the use of the hyphen in the amended annotations is that this form is very different phonetically from the "real" FPs described above in that it is a two-syllable vocalisation having a specific intonational form consisting of a "stylised" rise in pitch from relatively low, level F0 on the first syllable, to higher, but still level F0 on the second syllable. The hyphenated form *m-hm* was thought to better mirror the characteristics of this vocalisation, thus rendering the orthographic annotation more immediately transparent to the reader.

M-hm parallels neatly with informal renderings of *iva* 'yes' such as *ija* and *iwa*, as well as with the more frequent *eħe*, in having a significant "backchannelling" function (see Savino & Vella, forthcoming). Two further short expressions are annotated in the FP tier: *ta* and *ew*. The former, very common in everyday conversation, but only found four times in the data annotated, is described by Aquilina as "short for *taf*, you know", the latter as an "occasional variant of *jew*" (1990:1382; 1987:290). Although the status of both these vocalisations as "quasi-lexical" is unclear, they are mentioned here since they may share with *m-hm* the function of backchannelling mentioned earlier.

A third class of forms, in this case vocalisations which seem to be similar to the "ideophones and interjections" category listed by Borg & Azzopardi-Alexander (1997) also occur in the data. The latter include forms which have been annotated in our data as follows: *ffff*, *eqq*, *ħeqq* and *ttt*. The latter is described by Borg & Azzopardi-Alexander (1997:338) as an "alveolar click **ttt** [!]...commonly used to express lack of agreement with an interlocutor" and in fact it is this use that is attested in the data annotated, rather than a use indicating disapproval, often transcribed orthographically as *tsk*, and involving repetition of the click (see Mifsud & Borg, 1997).

3.1.4. Other tiers

The **TIs** tier indicates the presence, within the text, of Map Task target items. Target items included in the Map Task allow comparability across speakers and contain solely sonorant elements to allow for better pitch tracking e.g. *l-Amo**RIN*** in *Sqaq l-Amorin* 'Budgerigar Street', *l-Ewwel ta' **MEJ**ju* in *Triq l-Ewwel ta' Mejju* 'First of May Street' and ***A**mery* in *Triq Amery* 'Amery Street'. Target items were carefully selected to represent different syllable structure and stress possibilities in Maltese (see Vella & Farrugia 2006). The **TIs** tier is extremely useful to the phonetician using PRAAT as her/his main analysis tool since it increases searchability. It is not of great importance to the computational linguist, however, and will therefore not be considered further here.

As the tier name suggests, **MISC** contains miscellaneous information of different sorts. One example of a feature which gets recorded in the **MISC** tier is the case of inaudibly released final stops. Words ending in such plosives have been transcribed in standard orthography, a note also being added in the **MISC** tier to say that a final plosive had been inaudibly released. Cases of vowel coalescence, particularly at word boundaries, are also transcribed as in standard orthography, a note once more being inserted in the **MISC** tier to this effect. Other features noted in this tier include unexpected vowel quality realisations and idiosyncratic pronunciations including unusual use of aspiration, devoicing etc.; various "normal dysfluency"-type phenomena such as interruptions, abandoning of words, trailing off, unclear stretches of speech; also voice quality features such as creak and non-linguistic elements such as noise.

The contents of this tier are transcriber-dependent to an extent which is not fully desirable. Some general observations on dealing with features such as those in the **MISC** tier will be made in Section 4.

[4] It should be noted however that there may be an idiosyncratic element to these particular forms given that all instances of *ehh*s and *ehm*s noted in the data come from the same speaker.

Only rudimentary guidelines are available to date in the case of the three prosodic tiers **Tone**, **Prom** and **Functions**, for which the following outline will suffice. The annotation in these tiers is intended as a means of furthering research on Maltese prosody. Issues such as the relationship between perceived stress and intonation and the nature of the intonation patterns typical of Maltese and their distribution relative to discourse functions such as those involved in initiating a conversation, issuing an instruction, etc.

An important aim of this analysis is that of developing an adaptation for Maltese of the *Tone and Break Indices* (*ToBI*) framework for the annotation of prosody in the tradition of recent work in this area (see for example Silverman et al., 1992). Such an adaptation will impact on the development of standards for a Break Indices component, something which it is hoped the study of phonological boundaries and boundary strength mentioned above in 3.1.2 will in fact input into. A **B**(reak) **I**(ndices) tier does not in fact yet feature in the annotation carried out. Annotation of data using preliminary *ToBI*-style standards for Maltese based on the analysis of Maltese intonation carried out within the Autosegmental-Metrical framework (Pierrehumbert 1990; Ladd, 2008) by Vella (see, for example, 1995, 2003, 2007, 2009a, 2009b) should also contribute to further consolidation of the phonological analysis of Maltese prosody.

Typically, annotation begins in the **Prom** tier, to identify perceived prominence of accented and/or stressed syllables. With a stretch of speech thus highlighted as important in some way, related intonation patterns on the **Tone** tier, as well as discourse features on the **Functions** tier can then be annotated. The decision to include a **Prom** tier is based on work by Grabe (2001), on the annotation of the *IViE* (*Intonational Variation in English*) corpus, which specifies how pitch movement is "anchored" to syllables marked as prominent. The **Tone** tier then describes tones in terms of the way these link to identified prominent syllables and boundaries. This feature of the annotations should prove useful in the case of Maltese since a distinction between different degrees of prominence at the **Prom** tier may make it possible to account not only for more common "pitch accent"-type phenomena (see, e.g., Bolinger, 1958), but also for phenomena of the so-called "phrase accent"-type identified for Maltese (see Vella, 2003; following Grice, Arvaniti & Ladd 2000). Annotation at the level of prosody is currently underway.

A further tier, the **Functions** tier, contains information relating to discourse features as detailed by Carletta et al. (1995) in the coding of the HCRC Map Task Corpus. Their system describes typical features of turn-taking in conversation such as "initiating moves" like INSTRUCT or EXPLAIN and "response moves" like ACKNOWLEDGE or CLARIFY.

3.3 Alignment of tiers

As mentioned earlier in this Section (see Subsection 3.1.1) an important feature of PRAAT-style annotations is that involving the time-alignment of the waveform information to information in other tiers. Thus, the orthographic annotation of the spoken data goes hand in hand with the word-by-word segmentation in such a way that also allows information such as the starting time and ending time, and consequently the duration of each "segment" to be captured. Thus, the information in the SP1 and SP2 tiers in particular, but more generally that in the separate tiers of the annotation, involves time-alignment either of particular intervals or of particular points in the waveform to the information in other tiers of the annotation. This information, which is viewed in PRAAT as shown in Figure 1, is an extremely useful feature for the purposes of analysis. However, it poses a number of problems when it comes to incorporating the information from PRAAT TextGrid annotations into a corpus composed mainly of texts of a written form and it is this issue which will be discussed in the next Section.

4. Preparing SPeech ANnotations for integration into corpora of Maltese

As mentioned above, TextGrid annotations, whilst useful to phoneticians, do not necessarily allow for straightforward incorporation into corpora consisting mainly of written texts. The annotations, though stored in .txt format, contain information which is as such "redundant" for corpus linguistics. The TextGrid information relevant to the short excerpt shown in Figure 1 has been extracted from the relevant TextGrid and presented in the Appendix. The information entered into each interval labelled is listed together with an indication of start time and end time by tier. In the case of point tiers – such as the **Tone** tier in our annotations, which is however not illustrated in Figure 1 – any label inputted into the TextGrid is listed together with its position in time.

Having produced the TextGrid annotations using PRAAT, it was considered necessary to establish procedures for exporting the information of relevance for the incorporation of samples of spoken Maltese in a predominantly written corpus of Maltese. The desired outcome is machine-readable text containing not only the orthographic transcriptions relevant to the contributions of the speakers in the dialogue in the form of a playscript, but also any information from the PRAAT annotations which would be useful for processing the spoken "texts" in line with principles established also for the written ones. For ease of reference, a conventional playscript-type transcript of the excerpt shown in Figure 1 is given below:

SP1: M-hm.
SP2: M-hm?
SP1: (Overlapping) Għaddi minn bej' Sqaq il-Merill...u Triq Marmarà. Ibqa' tielgħa.
(Go between Merill Alley...and Marmarà Street. Keep moving upwards.)

Using a PRAAT script called GetLabelsOfIntervals_WithPauses (Lennes), it is possible to reduce the information shown in the Appendix as in Table 2 below:

Speaker	Words in sequence	Pause duration
		(0.23 s)
SP1:	M-hm.	
		(0.22 s)
SP2:	M-hm?	
		(-0.11s)
SP1:	Għaddi	
	Minn	
	bej'	
	Sqaq	
	il-Merill	
		(0.20s)
	U	
	Triq	
	Marmarà.	
		(0.24s)
	Ibqa'	
	tielgħa.	

Table 2: Contents of .txt file following extraction of information from PRAAT TextGrid

Exporting selected information from the PRAAT TextGrids as shown above makes it possible for some important features of the annotation carried out to be retained in text which, unlike TextGrids in their raw form, is easy to incorporate into a corpus composed mainly of written texts. Although, improvements on this preliminary attempt at exporting the data can be envisaged, the output of the script used can already be seen to contain a number of useful features. One of these is the fact that corpus-processing tasks such as paragraph (the spoken equivalent of which would be the utterance) and sentence splitting, as well as tokenisation, would seem to be relatively straightforward tasks given the word-by-word segmentation and annotation in the original format. POS tagging would need to proceed on the same lines as in the case of written texts.

A second very important feature of the output of the script is that it captures information on both pause, and on overlap, a very significant feature of speech which is completely absent from written texts. These two features are recorded in the right hand column of the output, a positive value indicating a pause, a negative value overlap. It should be a relatively straightforward for a script to be developed which will allow for such information to be converted into the appropriate tags.

5. The encoding of features of spoken Maltese

Given conversion of PRAAT annotations in a way similar to that described in Section 4 above, there remain few obstacles to overcome. Assuming some kind of mark-up similar to that used in the BNC, and an element <u> (utterance) corresponding to the written text <p> (paragraph) element, grouping a sequence of <s> (sentence) elements (BNC User Reference Guide), the short dialogue above could be encoded as follows:

```
<u who="SP1">                          <w..."bejn">
    <s...>                             <w..."Sqaq">
        <pause dur=0.23s>              <w..."il-Merill">
        <w..."M-hm">                   <pause dur=0.20s>
        <c c5="PUN">.</c>              <w..."u">
<u who="SP2">                          <w..."Triq">
        <w..."M-hm">                   <w..."Marmarà">
        <c c5="PUN">?</c>              <c c5="PUN">.</c>
<u who="SP1">                      <s...>
    <s...>                             <pause dur=0.24s>
        <overlap dur=-0.11s>           <w..."Ibqa'">
        <w..."Għaddi">                 <w..."tielgħa">
        <w..."minn">                   <c c5="PUN">.</c>
```

The above demonstrates that the output of the script used here already goes a long way towards helping us accomplish our purpose.

Some of the elements in the text, e.g. use of <...> to indicate elision could easily be adapted for the purposes of automatic tagging – the BNC suggests the use of an element <trunc> in such cases. Information relating to other elements such as <unclear> (entered in the **MISC** tier in the current TextGrid annotations), could also be retrieved, albeit possibly in a less straightforward fashion.

One element of particular interest in the context of this paper is the element <vocal>. The BNC User Reference Guide describes this as: "(Vocalized semi-lexical) any vocalized but not necessarily lexical phenomenon for example voiced pauses, non-lexical backchannels, etc.". One of the outcomes of the project *SPAN* is in fact a categorisation of different types of vocalisations as follows (see Vella et al., 2009):

1. "real" FPs such as *eee*, *mmm* and *emm* having an actual pause as their counterpart;
2. non-lexical vocalisations such as *m-hm* which parallel with quasi-lexical vocalisations such as *eħe* and with lexical words such as *iva*; and
3. "paralinguistic vocalisations" such as *fff*, *ħeqq*, *ttt* etc.

Given that, in all cases, the elements in these categories consist of a relatively closed set of items – which after all could be added to in cases of new items being identified – such elements should be relatively easy to identify on the basis of their orthographic rendering in the annotations, although one would need to assume proper training of transcribers to established standards and guidelines. It is being suggested here that vocalisations involving some

element of meaning would be better tagged as "words", thus leaving the element "vocalisations" as a means of recording events of a purely non-linguistic nature. A full categorisation of different events of this sort, as well as of other vocalisations of a "quasi-lexical" nature still awaits research.

6. Conclusion

In conclusion, standards and guidelines for the orthographic annotation, as well as preliminary standards for the annotation of prosody, of spoken Maltese, are in place. Exportation of TextGrid information to a format more readily incorporable into corpora of written data is also doable. However, possibilities for automating or semi-automating procedures of conversion need to be explored. Lastly, improved knowledge of the workings of these relatively-less-described features of Maltese should serve not only to improve the quality of HLTs such as Text-to-Speech systems for Maltese, but also to improve methodologies for the evaluation of such HLTs.

7. Acknowledgements

We would like to thank the University of Malta's Research Fund Committee (vote numbers 73-759 and 31-418) for making funding for the projects *SPAN* (1) and *SPAN* (2) available for the years starting January 2007 and January 2008.

8. References

Akkademja tal-Malti. (2004). *Tagħrif fuq il-Kitba Maltija II*. Malta: Klabb Kotba Maltin.

Aquilina, J. (1987/1990). *Maltese-English Dictionary.* Volumes 1 & 2. Malta: Midsea Books Ltd.

Boersma, P., Weenick, D. (2008). PRAAT: doing phonetics by computer. (Version 5.0.08). http://www.praat.org visited 11-Feb-08.

Bolinger, D. (1958). A theory of pitch accent in English. *Word*, 14, pp. 109--149.

Borg, A., Azzopardi-Alexander, M. (1997). *Maltese*. [Descriptive Grammars]. London/New York: Routledge.

British National Corpus, http://www.natcorp.ox.ac.uk/corpus/index.xml visited 20-Feb-10.

Carletta, J., Isard, A., Kowtko, J., Doherty-Sneddon, G., Anderson, A. (1995). The coding of dialogue structure in a corpus. In J.A. Andernach, S.P. van de Burat & G.F. van der Hoeven (Eds.), *Proceedings of the Twentieth Workshop on Language Technology: corpus-based approaches to dialogue modelling*, pp. 25--34.

Cruttenden, A. (1997). *Intonation*. 2nd edition. Cambridge: Cambridge University Press.

Dalli, A. (2001). Interoperable extensible linguistic databases. In *Proceedings of the IRCS Workshop on Linguistics Databases*, University of Pennsylvania, Philadelphia, pp. 74--81.

Gibbon, D., Moore, R., Winski, R. (1997). *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter.

Grabe, E. (2001). The IViE Labelling Guide. (Version 3). http://www.phon.ox.ac.uk/files/apps/IViE//guide.html# visited 12-Apr-10

Grice, M., Ladd, D.R.L., Arvaniti, A. (2000). On the place of phrase accents in intonational phonology. *Phonology*, 17, pp. 143--185.

HCRC Map Task Corpus, http://www.hcrc.ed.ac.uk/maptask/ visited 10-Nov-09.

Kunsill tal-Malti, http://kunsilltalmalti.gov.mt/filebank/documents/Decizjonijiet1_25.07.08.pdf visited 15-Apr-10.

Ladd, D.R.L. (2008). *Intonational Phonology*. 2nd edition. Cambridge: Cambridge University Press.

Lennes, M. (2010). Mietta's Praat scripts, GetLabelsOfIntervals_WithPauses script. http://www.helsinki.fi/~lennes/praat-scripts/ visited 18–Jan-10.

Mifsud, M., Borg, A. (1997). *Fuq l-Għatba tal-Malti*. [Threshold Level in Maltese]. Strasbourg: Council of Europe Publishing.

Pierrehumbert, J. (1980). The Phonetics and Phonology of English Intonation. Ph.D. thesis, MIT.

Rosner, M., Caruana, J., Fabri, R. (1998). *MaltiLex*: a computational lexicon for Maltese. In *Proceedings of the COLING-ACL Workshop on Computational Approaches to Semitic Languages.* Morristown, NJ: Association for Computational Linguistics, pp. 97--101.

Rosner, M., Caruana, J., Fabri, R., Loughraieb, M., Montebello, M., Galea, D., Mangion, G. (2000). Linguistic and computational aspects of *MaltiLex*. In *Proceedings of ATLAS: The Arabic Translation and Localization Symposium.* PLACE, pp. 2 -- 9.

Rosner, M. (2009). Electronic language resources for Maltese. In B. Comrie, R. Fabri, E. Hume, M. Mifsud & M. Vanhove (Eds.), *Introducing Maltese Linguistics.* Amsterdam: John Benjamins, pp. 251--276.

Savino, M., Vella, A. (Forthcoming). Intonational backchannelling strategies in Italian and Maltese Map Task dialogues.

Shriberg, E.E. (1999). Phonetic consequences of speech disfluency. In *Proceedings of the 13th International Congress of Phonetic Sciences 1999,* San Francisco, CA, pp. 619--622.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrrehumbert, J., Hirschberg, J. (1992). ToBI: a standard for labeling English prosody. In *Proceedings of the 1992 International Conference on Spoken Language Processing*. Banff, Canada, p. 867--870.

Vella, A. (1995). Prosodic Structure and Intonation in Maltese and its Influence on Maltese English. Unpublished Ph.D thesis, University of Edinburgh.

Vella, A. (2003). Phrase accents in Maltese: distribution and realisation. In *Proceedings of the 15th International Congress of Phonetic Sciences*. Barcelona, pp. 1775--1778.

Vella, A. (2007). The phonetics and phonology of *wh*-question intonation in Maltese. In *Proceedings of the 16th International Congress of Phonetic Sciences*. Saarbrücken, pp. 1285--1288.

Vella, A. (2009a). Maltese intonation and focus structure. In R. Fabri (Ed.), *Maltese Linguistics: A Snapshot. In Memory of Joseph A. Cremona.* [Il-Lingwa Tagħna Vol. 1]. Bochum: Niemeyer, pp. 63--92.

Vella, A. (2009b). On Maltese prosody. In B. Comrie, R. Fabri, E. Hume, M. Mifsud & M. Vanhove (Eds.), *Introducing Maltese Linguistics*. Amsterdam: John Benjamins, pp. 47--68.

Vella, A., Farrugia, P-J. (2006). *MalToBI* – building an annotated

corpus of spoken Maltese. In *Proceedings of Speech Prosody 2006*, Dresden.

Vella, A., Chetcuti, F., Grech, S., Spagnol, M. (2008). SPeech ANnotation: developing guidelines for spoken corpora. Paper given at the *1st International Conference of Maltese Linguistics*, Bremen.

Vella, A., Chetcuti, F., Grech, S., Spagnol, M. (2009). Interspeaker variation in the type and prosody of filled pauses in Maltese. Paper given at the *2nd International Conference of Maltese Linguistics*, Bremen.

Yvnge, V. (1970). On getting a word in edgewise. *Papers from the Sixth Regional Meeting, Chicago Linguistic Society*, Chicago, pp. 567-577.

Appendix

TextGrid information for sample excerpt in Figure 1. The sequences of left and right angled brackets indicate the positions at which information from the different tiers of the original TextGrid was removed in order to make it possible for time-aligned information from the various tiers to be presented here.

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 225.7226530612245
tiers? <exists>
size = 6
item []:
    item [1]:
        class = "IntervalTier"
        name = "SP1"
<<<<<<<<<<>>>>>>>>>>
intervals [24]:
            xmin = 18.943357990906726
            xmax = 19.261373640829614
            text = "M-hm."
        intervals [25]:
            xmin = 19.261373640829614
            xmax = 19.643059096210124
            text = ""
        intervals [26]:
            xmin = 19.643059096210124
            xmax = 19.89371820123613
            text = "Għaddi"
        intervals [27]:
            xmin = 19.89371820123613
            xmax = 20.065239076749062
            text = "minn"
        intervals [28]:
            xmin = 20.065239076749062
            xmax = 20.231585573720867
            text = "bej"
        intervals [29]:
            xmin = 20.231585573720867
            xmax = 20.49279384475206
            text = "Sqaq"
        intervals [30]:
            xmin = 20.49279384475206
            xmax = 21.030082586632584
            text = "il-Merill"
        intervals [31]:
            xmin = 21.030082586632584
            xmax = 21.225530013079816
            text = ""
        intervals [32]:
            xmin = 21.225530013079816
            xmax = 21.338326610341518
            text = "u"
        intervals [33]:
            xmin = 21.338326610341518
            xmax = 21.521897831205
            text = "Triq"
        intervals [34]:
            xmin = 21.521897831205
            xmax = 22.091024710212217
            text = "Marmar\a`."
        intervals [35]:
            xmin = 22.091024710212217
            xmax = 22.335863155310836
            text = ""
        intervals [36]:
            xmin = 22.335863155310836
            xmax = 22.584860481468173
            text = "Ibqa'"
        intervals [37]:
            xmin = 22.584860481468173
            xmax = 22.997308645192785
            text = "tielgħa."
<<<<<<<<<<>>>>>>>>>>
item [2]:
        class = "IntervalTier"
        name = "SP2"
<<<<<<<<<<>>>>>>>>>>
intervals [8]:
            xmin = 18.71206799854182
            xmax = 19.48013067794322
            text = ""
        intervals [9]:
            xmin = 19.48013067794322
            xmax = 19.756995053040125
            text = "M-hm?"
        intervals [10]:
            xmin = 19.756995053040125
            xmax = 23.648962491299642
            text = ""
        intervals [11]:
            xmin = 23.648962491299642
            xmax = 24.163953016171252
            text = "Sqaq"
        intervals [12]:
            xmin = 24.163953016171252
            xmax = 24.812517158406998
            text = "il-Merill"
        intervals [13]:
            xmin = 24.812517158406998
            xmax = 24.9184775982589
            text = ""
        intervals [14]:
xmin = 24.9184775982589
            xmax = 24.988121382607528
            text = "u"
        intervals [15]:
            xmin = 24.988121382607528
            xmax = 25.060897544296406
            text = ""
        intervals [16]:
            xmin = 25.060897544296406
            xmax = 25.539234518515816
            text = "Triq"
        intervals [17]:
            xmin = 25.539234518515816
            xmax = 26.066892216144574
            text = ""
        intervals [18]:
            xmin = 26.066892216144574
            xmax = 26.63328825195686
            text = "Marmara`."
<<<<<<<<<<>>>>>>>>>>
item [3]:
        class = "IntervalTier"
        name = "Br-Pa-Os"
<<<<<<<<<<>>>>>>>>>>
intervals [21]:
            xmin = 18.71206799854182
            xmax = 18.943357990906726
            text = "Pause-C"
        intervals [22]:
            xmin = 18.943357990906726
            xmax = 19.643059096210124
            text = " "
        intervals [23]:
            xmin = 19.643059096210124
            xmax = 19.756995053040125
            text = "Overlap"
        intervals [24]:
            xmin = 19.756995053040125
            xmax = 21.030082586632584
            text = ""
        intervals [25]:
            xmin = 21.030082586632584
            xmax = 21.225530013079816
            text = "Break"
        intervals [26]:
            xmin = 21.225530013079816
            xmax = 22.091024710212217
            text = ""
        intervals [27]:
xmin = 22.091024710212217
            xmax = 22.335863155310836
            text = "Pause"
<<<<<<<<<<>>>>>>>>>>
item [4]:
        class = "TextTier"
        name = "TIs"
<<<<<<<<<<>>>>>>>>>>
        points [7]:
            time = 20.231585573720867
            mark = "TB8"
        points [8]:
            time = 21.030082586632584
            mark = "TE8"
        points [9]:
            time = 21.338326610341518
            mark = "TB17"
        points [10]:
            time = 22.091024710212217
            mark = "TE17"
<<<<<<<<<<>>>>>>>>>>
item [5]:
        class = "IntervalTier"
        name = "FPs"
<<<<<<<<<<>>>>>>>>>>
        intervals [2]:
            xmin = 18.943357990906726
            xmax = 19.261373640829614
            text = "FP"
        intervals [3]:
            xmin = 19.261373640829614
            xmax = 19.48013067794322
            text = ""
        intervals [4]:
            xmin = 19.48013067794322
            xmax = 19.756995053040125
            text = "FP"
<<<<<<<<<<>>>>>>>>>>
item [6]:
        class = "IntervalTier"
        name = "MISC"
<<<<<<<<<<>>>>>>>>>>
        intervals [4]:
            xmin = 19.261373640829614
            xmax = 19.48013067794322
            text = "Aspiration"
```

A Web Application for Dialectal Arabic Text Annotation

Yassine Benajiba and Mona Diab

Center for Computational Learning Systems
Columbia University, NY, NY 10115
{ybenajiba,mdiab}@ccls.columbia.edu

Abstract

Design and implementation of an application which allows many annotators to annotate data and enter the information into a central database is not a trivial task. Such an application has to guarantee a high level of security, consistent and robust back-ups for the underlying database, and aid in increasing the speed and efficiency of the annotation by providing the annotators with intuitive GUIs. Moreover it needs to ensure that the data is stored with a minimal amount of redundancy in order to simultaneously save all the information while not losing on speed. In this paper, we describe a web application which is used to annotate many Dialectal Arabic texts. It aims at optimizing speed, accuracy and efficiency while maintaining the security and integrity of the data.

1. Introduction

Arabic is spoken by more than 300 million people in the world, most of them live in Arab countries. However the form of the spoken language varies distinctly from the written standard form. This phenomenon is referred to as diglossia (Ferguson, 1959). The spoken form is dialectal Arabic (DA) while the standard form is modern standard Arabic (MSA). MSA is the language of education in the Arab world and it is the language used in formal settings, people vary in their degree of proficiency in MSA, however it is not the native tongue of any Arab. MSA is shared across the Arab world. DA, on the other hand, is the everyday language used in spoken communication and is emerging as the form of Arabic used in web communications (social media) such as blogs, emails, chats and SMS. DA is a pervasive form of the Arabic language, especially given the ubiquity of the web.

DA varies significantly from region to region and it varies also within a single country/city depending on so many factors including education, social class, gender and religion. But of more relevance to our object of study, from a natural language processing (NLP) perspective, DA varieties vary significantly from MSA which poses a serious impediment for processing DA with tools designed for MSA. The fact is that most of the robust tools designed for the processing of Arabic to date are tailored to MSA due to the abundance of resources for that variant of Arabic. In fact, applying NLP tools designed for MSA directly to DA yields significantly lower performance (Habash et al., 2008; Benajiba et al., 2008) making it imperative to direct research to building resources and dedicated tools for DA processing.

DA lack large amounts of consistent data due to several factors: the lack of orthographic standards for the dialects, the lack of overall Arabic content on the web, let alone DA content. Accordingly there is a severe deficiency in the availability of computational annotations of DA data.

Any serious attempt at processing real Arabic has to account for the dialects. Even broadcast news which is supposed to be MSA has non-trivial DA infiltrations. In broadcast news and talk shows, for instance, speakers tend to *code switch* between MSA and DA quite frequently. Figure 1 illustrates an example taken from a talk show on Aljazeera[1] where the speaker explains the situation of women who are film makers in the Arab world. The DA word sequences are circled in red and the rest of the text is in MSA. In (Habash et al., 2008), the authors show that in broadcast conversation, DA data represents 72.3% of the text. In weblogs, the amount of DA is even higher depending on the domain/topic of discussion where the entire text could be written in DA.

Language used in social media pose a challenge for NLP tools in general in any language due the difference in genre. Social media language is more akin to speech in nature and people tend to be more loose in their writing standards. The challenge arises from the fact that the language is less controlled and more speech like where many of the textually oriented NLP techniques are tailored to processing edited text. The problem is exacerbated for Arabic writing found on the web because of the use of DA in these genres. DA writing lacks orthographic standards, on top of the other typical problems associated with web media language in general of typographical errors and lack of punctuation. Figure 2 shows a fully DA text taken from an Arabic weblog[2].

Our Cross Lingual Arabic Blog Alerts (COLABA) project aims at addressing these gaps both on the resource creation level and the building of DA processing tools. In order to achieve this goal, the very first phase consists of gathering the necessary data to model:

- Orthographic cleaning and punctuation restoration (mainly sentence splitting);
- Dialect Annotation;
- Lemma Creation;
- Morphological Profile Creation.

Across all these tasks, we have designed a new phonetic scheme to render the DA in a conventionalized internal orthographic form details of which are listed in (Diab et al., 2010b). We believe that creating a repository of

[1] http://www.aljazeera.net/

[2] $http://www.paldf.net/forum/$

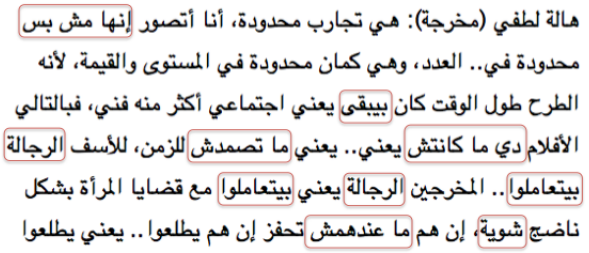

Figure 1: An illustrating example of MSA - DA code switching.

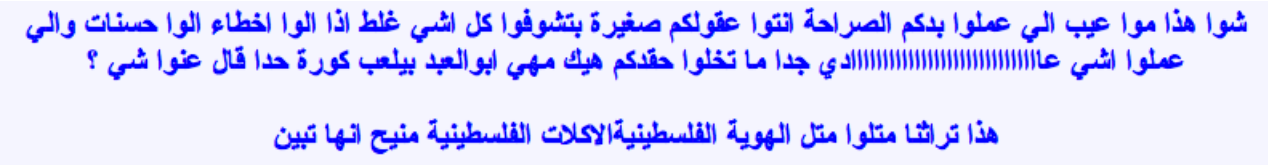

Figure 2: An illustrating example of a DA text on a weblog.

consistent annotated resources allows for the building of applications such as Information Retrieval, Information Extraction and Statistical Machine Translation on the DA data. In this project, we have targeted four Arabic Dialects, namely: Egyptian, Iraqi, Levantine, and Moroccan. And the harvested data is on the order of half a million Arabic blogs. The DA data is harvested based on manually created queries in the respective dialects as well as a list of compiled dialect specific URLs. Once the data is harvested it is automatically cleaned from metadata and the content part is prepared for manual annotation.

The application that we present in this paper, COLANN_GUI, is designed and implemented in the framework of the COLABA project.

COLANN_GUI is the interface used by the annotators to annotate the data with the relevant information. COLANN_GUI uses two different servers for its front-end and back-end components. It also allows many annotators to access the database remotely. It offers several views depending on the type of user and the annotation task assigned to an annotator at any given time. The decision to develop an annotation application in-house was taken after unsuccessfully trying to find an off-the-shelf tool which can offer the functionalities we are interested in. Some of these functionalities are:

- Task dependency management: Some of the annotation tasks are dependent on each other whereas others are completely detached. It is pretty important in our tasks to be able to manage the annotation tasks in a way to keep track of each word in each sentence and organize the information entered by the annotator efficiently. It is conceivable that the same word could have different annotations assigned by different annotators in different tasks whereas most the available tools do not have the flexibility to be tailored is such fashion; and
- Annotators' management: the tool should be able to allow the lead annotators to assign different tasks to different annotators at different times, help them trace the annotations already accomplished, and should allow them to give illustrative constructive feedback from within the tool with regards to the annotation quality.

Even though many of these annotation tools, such as GATE(Damljanovic et al., 2008; Maynard, 2008; Aswani and Gaizauskas, 2009), Annotea(Kahan et al., 2001) and MnM(Vargas-Vera et al., 2002) among others, have proven successful in serving their intended purposes, none of them was flexible enough for being tailored to the COLABA goals.

The remainder of this paper is organized as follows: We give an overview of the system in Section 2.; Section 3. illustrates the detailed functionalities of the application; Section 4. describes each of the annotation tasks handled by the application; We give further details about the database in Section 5. and finally, some future directions are shared in Section 6..

2. Overall System View

COLANN_GUI is a web application. We have chosen such a set up, in lieu of a desktop one, as it allows us to build a machine and platform independent application. Moreover, the administrator (or super user) will have to handle only one central database that is multi-user compatible. Furthermore, the COLANN_GUI is browser independent, i.e. all the scripts running in the background are completely browser independent hence allowing all the complicated operations to run on the server side only. COLANN_GUI uses PHP scripts to interact with the server database, and uses JavaScripts to increase GUI interactivity.

Safety and security are essential issues to be thought of when designing a web application. For safety considerations, we employ a subversion network (SVN) and auto-

matic back-up servers. For security considerations we organize our application in two different servers, both of which is behind several firewalls (see Figure 3).

3. COLANN_GUI: A Web Application

As an annotation tool, we have designed COLANN_GUI with three types of users in mind: Annotators, Lead Annotators, and Super User. The design structure of COLANN_GUI aims to ensure that each annotator is working on the right data at the right time. The Super User and Lead Annotator views allow for the handling of organizational tasks such as database manipulations, management of the annotators as well as control of in/out data operations.
Accordingly, each of these different views is associated with different types of permissions which connect to the application.

3.1. Super User View

The Super User has the following functionalities:

1. Create, edit and delete tables in the database
2. Create, edit and delete lead accounts
3. Create, edit and delete annotator accounts
4. Check the status of the annotation tasks for each annotator
5. Transfer the data which needs to be annotated from text files to the database
6. Generate reports and statistics on the underlying database
7. Write the annotated data into XML files

3.2. Lead Annotator View

The Lead Annotator view shares points 3 and 4 of the Super User view. In addition, this view has the following additional functionalities:

1. Assign tasks to the annotators
2. Check the annotations submitted by the annotators
3. Communicate annotation errors to the annotators
4. Create gold annotations for samples of the assignment tasks for evaluation purposes. Their annotations are saved as those of a special annotator
5. Generate inter-annotator agreement reports and other types of relevant statistics on the task and annotator levels

3.3. Annotator View

The annotator view has the following functionalities:

1. Check status of his/her own annotations
2. Annotate the assigned units of data
3. Check the overall stats of other annotators' work for comparative purposes
4. An annotator could check the speed of others (anonymously and randomized) on a specific task once they submit their own
5. View annotations shared with them by the Lead Annotator

4. Annotation Tasks

A detailed description of the annotation guidelines goes beyond the scope of this paper. The annotation guidelines are described in detail in (Diab et al., 2010b). We enumerate the different annotation tasks which our application provides. All the annotation tasks can only be performed by a user of category *Annotator* or *Lead Annotator* for the creation of the gold evaluation data. In all the tasks, the annotator is asked to either save the annotation work, or submit it. If saved they can go back and edit their annotation at a later time. Once the work is submitted, they are not allowed to go back and edit it. Moreover, the annotators always have direct access to the relevant task guidelines from the web interface by pressing on the information button provided with each task.
The annotation tasks are described briefly as follows:

1. *Typo Identification and Classification and Sentence Boundary Detection:* The annotator is presented with the raw data as it is cleaned from the meta data but as it would have been present on web. Blog data is known to have all kinds of speech effects and typos in addition to a severe lack of punctuation.
   Accordingly, the first step in content annotation is to identify the typos and have them classified and fixed, in addition have sentence boundaries identified.

   The typos include: (i) gross misspellings: it is recognized that DA has no standard orthography, however many of the words are cognates/homographs with MSA, the annotator is required to fix misspelling of such words if they are misspelled for example المساجذ *AlmsAj*, "the mosques" would be fixed and re-entered as المساجد *AlmsAjd*; (ii) speech effects: which consists of rendering words such as "Goaaaal" to "Goal"; and (iii) missing spaces. The annotator is also asked to specify the kind of typo found. Figure 4 shows a case where the annotator is fixing a "missing space" typo.
   The following step is sentence boundary detection. This step is crucial for many of the language tools which cannot handle very long sequences of text, e.g. syntactic parsers. In order to increase the speed and efficiency of the annotation, we make it possible to indicate a sentence boundary by clicking on a word in the running text. The sequence of words is simply split at that click point. The annotator can also decide to merge two sequences of words by clicking at the beginning of a line and it automatically appends the current line to the previous one. It is worth noting that all the tasks that follow depend on this step being completed. Once this task is completed, the data is sent to a coarse grained level of dialect identification (DI) pipeline described in detail in (Diab et al., 2010a). The result of this DI process is the identification of the

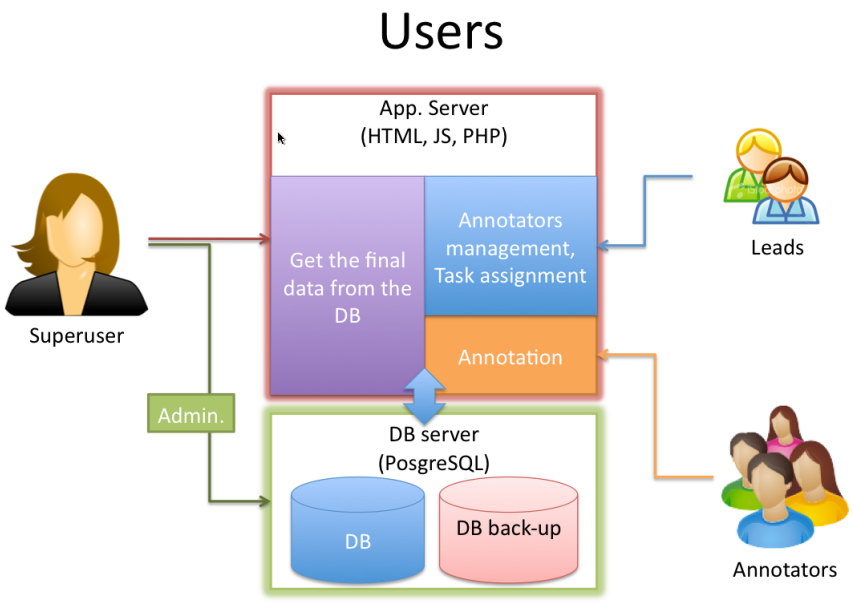

Figure 3: Servers and views organization.

Figure 4: Typo Identification and Fixing.

problem words and sequences that are not recognized by our MSA morphological analyzer, i.e. the words don't exist in our underlying dictionaries.[3]

2. *Dialect annotation:* For each word in the running text (after the content cleaning step mentioned before), the annotator is asked to specify its dialect(s) by picking from a drop down menu. Moreover they are requested to choose the word's level of dialectalness on a given scale. Finally, they are required to provide the phonetic transcription of word as specified in our guidelines on rendering DA in the COLABA Conventional Orthography (CCO).

The GUI at this point only allows the annotator to submit his/her annotation work when all the words in the text are annotated. The annotators are given the option to mark a word as unknown.

Another functionality that we have added in order to help the annotators speed up their annotation in an efficient way is a color coding system for similar words. If the annotator enters the possible dialects, relevant annotation, and the phonetic CCO transliteration for a surface word w_i. The annotated words change color

[3]It is important to note that we run the data through the morphological analyzer as opposed to matching against the underlying dictionary due to the fact the design decision we made early on that our dictionaries will have lemmas and rules associated with them rather than exhaustively listing all possible morphological forms which could easily be in the millions of entries.

to red. This allows the annotator to know which words have already been annotated by simply eye balling the words colored in red in the overall document undergoing annotation. Second, the script will look for all the words in the text which have the same surface form as w_i, i.e. all instances/occurrences of annotated w_i, and it will color each of these instances in blue. The annotator then, can simply skip annotating these words if s/he judges them to have the same annotation as the original, so it ends up being a revision rather than a new annotation. It is easy to understand how this simple change of color coding can facilitate the annotation job and increase the efficiency of the annotation process by an example. In a long Arabic blog text, frequent function words such as مش, *m$*, "not", will only need to be annotated once.

Figure 6 shows an illustrating example of the dialect annotation process via a screenshot of the task.

3. *Lemma Creation:* In this task, the annotators are asked to provide the underlying lemma forms (citation forms) for surface DA words. The lemmas constiture the dictionary entry forms in our lexical resources. The resource aims to have a large repository of DA lemmas and their MSA and English equivalents as well as DA example usages as observed in the blog data in the COLABA project. Accordingly, the annotator is provided with a surface DA word and instances of it's usage from example sentences in the blogs and they are required to provide the corresponding lemma, MSA equivalent, English equivalent, gross dialect id. Once they provide the lemma, they have to identify which example usage is associated with the lemma they created. All the lemma information is typed in using the CCO transcription scheme that COLABA specifies. It is worth noting that this task is completely independent from the Dialect Annotation task. Hence annotators could work directly on this task, i.e. after fixing typos and sentence boundaries are identified and the DI process is run.

   Accordingly, after the data undergoes the various clean up steps mentioned earlier, the data goes through the DI process as follows:

   (a) Transliterate the Arabic script of the blogs into the Buckwalter Transliteration scheme (Buckwalter, 2004) after the previous content clean up tasks of typo and sentence boundary handling. This process also identifies the foreign word character encoding if they exist in the text;

   (b) Use the DI pipeline to identify the DA words within each document;

   (c) Build a ranked list of all the surface DA words observed in the input document set based on their frequency of occurrence, while associating each surface word with the sentences in which it occurred in the document collection;

   Thus, we have grouped the DA words by surface form and used them as key entries in our database allowing us the ability to access them easily with their recurrent examples which are in turn identified uniquely by sentence number and document number. For instance, let us consider all the sentences where the surface word مركبه, *mrkbh*, appears in our data. For illustration in this paper, we provide the English translation and the Buckwalter transliteration, however in the actual interface the annotators only see the surface DA word and associated examples in Arabic script as they occur in the data, but after being cleaned up from meta data, html mark up, typos are fixed and sentence boundaries identified.

ها ها ها ها لو يعطوني بليار والله **مركبه** يماااااااه ...
Buckwalter Transliteration: *hA hA hA hA lw yEtwny blyAr wAllh mrkbh ymAAAAAAAAh ...*
English Gloss: hahaha even if they give me a billion I wouldn't **ride it** muuuuuum
...انجرف **مركبه** الى وادي ذيبان بسبّ ارتفاع
Buckwalter Transliteration: *A*njrf mrkbh AlY wAdy *ybAn bsbb ArtfAE ...
English Gloss: his boat drifted to the Dhibane valley because of the increase of the level ...
ده سامي مركبه فيلبلكونه
Buckwalter Transliteration: *d*h sAmy mrkbh fylblkwnh
English Gloss: Samy has it **s**et up in the balcony

   These sentences are shown to the annotator and s/he is asked to identify the number of lemmas for this surface word. For instance, in the second example, we find sentences where *mrkbh* appears as "his boat" and in the first example it appears as "ride". Accordingly in these examples, the annotator should indicate that there are three different lemmas for the surface form *mrkbh* rendered in CCO transliteration scheme as *rikib*, *markib*, and *merakib*, respectively.

   Figure 5 shows an illustrating example.

4. *Morphological Profile Creation:* Only for those words which have been already annotated with the lemma information in the previous step do we proceed for further annotation. For those lemmas in the database already, we add more detailed morphological information. The annotator is shown one lemma at a time with a set of example sentences where the surface form of the lemma is used. Thereafter the annotator is asked to select a part of speech tag (POS-tag). Figure 7 shows that when a POS-tag is selected the interface shows the type of information required accordingly.

In all these tasks, the application is always keeping track of the time that took each annotator for each task unit. Such information is necessary to compare speed and efficiency among annotators and also for the annotators themselves to be able to compare themselves to the best and the worst across a task.

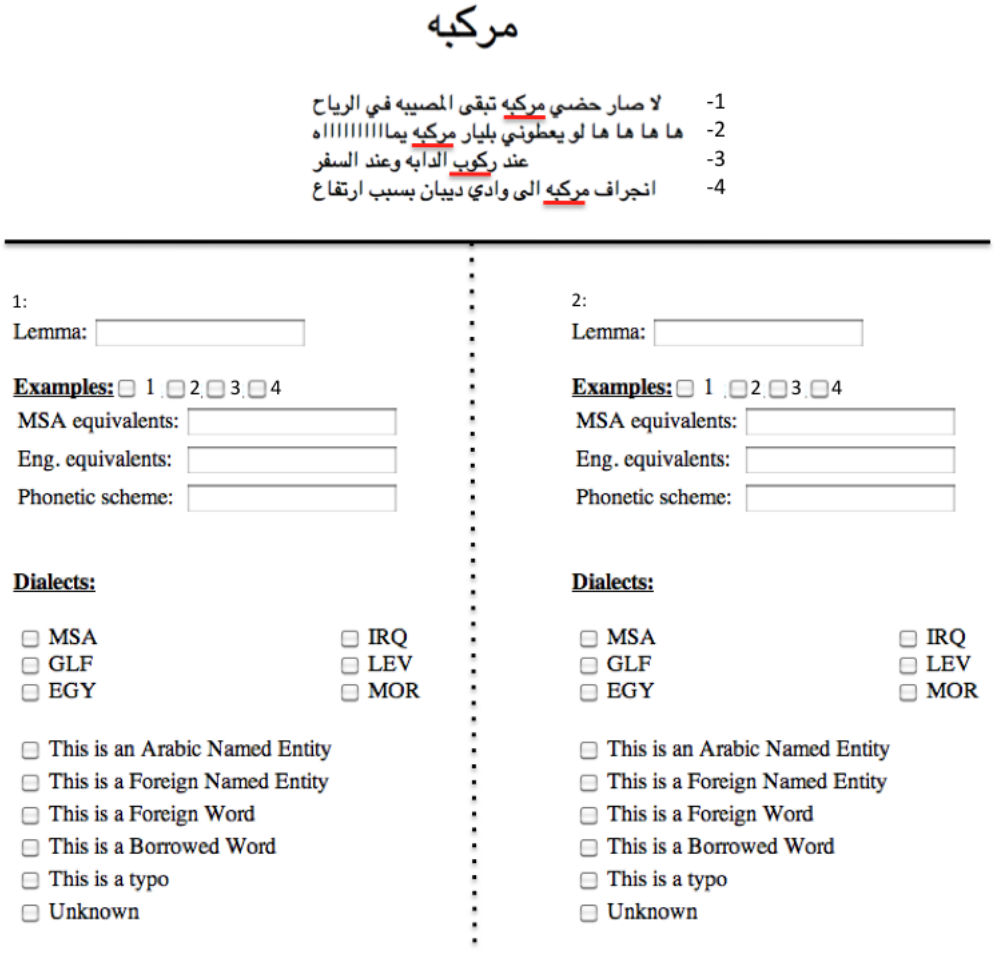

Figure 5: Illustrating example of lemma creation.

5. The Database

For this application we need a sophisticated database. We use the freely available `Postgresql` management system[4]. The database system is able to save all the annotations which we have described in Section 3.. But it also holds other information concerning the users and their annotation times. Our database contains 22 relational tables, which can be split into the following categories:

- *Basic information*: We have created a number of tables which save basic information that is necessary for all the tasks. By saving such information in the database our application becomes very easy to update and maintain. For instance, if we decide to add a new POS-tag we just have to add it in the appropriate table.
- *Annotation*: These tables are the core of the database. For each of the annotation tasks described in Section 4. it saves all the information entered by the annotator while keeping the redundancy of the information at a minimum. For instance, for each sentence in the data, we want to save all the information entered about the dialect, the lemmas and the morphological information of the dialectal words while saving only one instance of the actual sentence in only one table and relate all the annotation records to it. Only by doing so are we able to save information about millions of words in the database while keeping it easily and efficiently accessible. Finally, we also save the time (in seconds) taken by the annotators to complete the annotation tasks. This information is necessary for inter-annotator speed comparison.
- *Assignments*: These tables hold information about how many task assignments have been assigned to each annotator and how many of them have already been annotated and submitted. This is directly related to the assignment task of the lead annotators described in Subsection 3.2..
- *Users Permissions*: When a user is created, we assign a certain category to her/him. This information is used by all the scripts to decide on user privileges.
- *Connection*: Whenever a user is connected, this information is communicated to the database. By doing so we are able to deny a user to connect from two machines at the same time.

As mentioned in Section 1., our database is located on a separate server from the web server. This web server can

[4] http://www.postgresql.org/

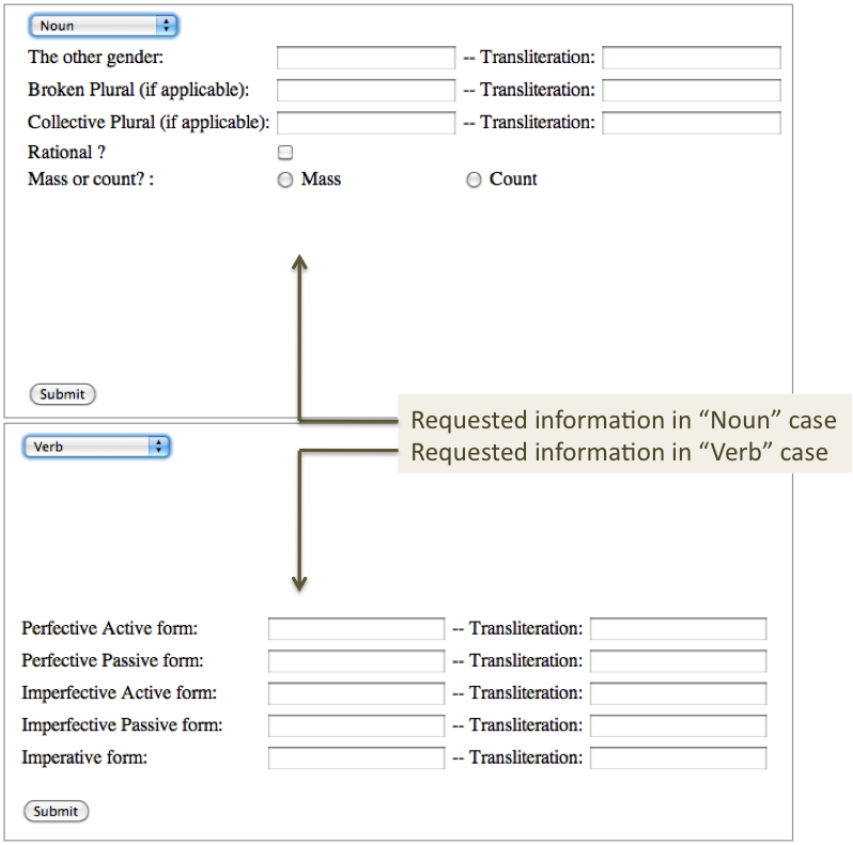

Figure 7: Illustrating example of the requested information when an annotator chooses the POS-tag Noun or Verb.

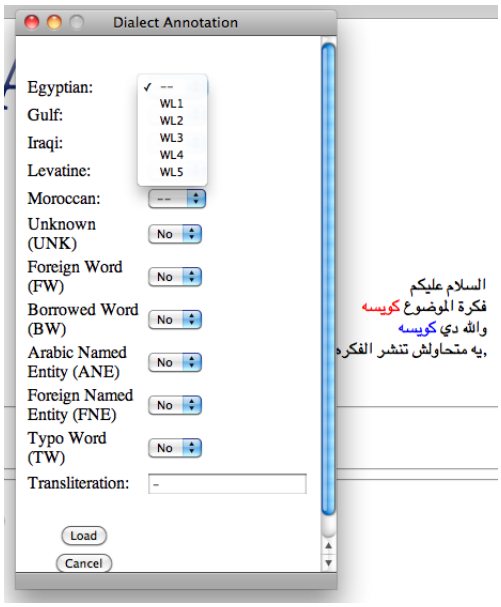

Figure 6: Illustrating example of dialect annotation.

send requests to the database server through an `ssh` tunnel which helps forward services between the two servers with encrypted data.[5]

6. Future Work

We are constantly updating our interface incorporating feedback from the annotators and lead annotators on the various tasks. The data that is annotated using our application is intended to build efficient models of four different dialects that cover all the major Arabic dialects. The models will be useful for several NLP applications:

- Automatic spelling correction;
- Automatic sentence boundary detection;
- Automatic dialect identification and annotation;
- Lemmatization and POS-tagging;
- Information Retrieval and Advanced search;
- Named Entity, foreign words and borrowed words detection;

However, it is not possible to aim at such advanced applications without a consistent annotation where the efficiency of the application which we describe in this paper plays a pivotal role.

Acknowledgments

This work has been funded by ACXIOM Corporation.

[5] http://www.ssh.com/support/documentation/online/ssh/adminguide/32/Port_Forwarding.html

7. References

N. Aswani and R. Gaizauskas. 2009. Evolving a general framework for text alignment: Case studies with two south asian languages. In *Proceedings of the International Conference on Machine Translation: Twenty-Five Years On*.

Y. Benajiba, M. Diab, and P. Rosso. 2008. Arabic named entity recognition using optimized feature sets. In *Proceedings of EMNLP'08*, pages 284–293.

T. Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0.

D. Damljanovic, V. Tablan, and K. Bontcheva. 2008. A Text-based Query Interface to owl Ontologies. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC)*.

M. Diab, N. Habash, O. Rambow, M. AlTantawy, and Y. Benajiba. 2010a. COLABA: Arabic Dialect Annotation and Processing. In *Proceedings of the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages at LREC*.

Mona Diab, Nizar Habash, Reem Faraj, and May Ahmar. 2010b. Guidelines for the Annotation of Dialectal Arabic. In *Proceedings of the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages at LREC*.

C. A. Ferguson. 1959. Diglossia. *Word*, 15:325–340.

N. Habash, O. Rambow, M. Diab, and R. Kanjawi-Faraj. 2008. Guidelines for Annotation of Arabic Dialectness. In *Proceedings of the LREC Workshop on HLT & NLP within the Arabic world*.

J. Kahan, M.R. Koivunen, E. Prud'Hommeaux, and R.R. Swick. 2001. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the WWW10 Conference*.

D. Maynard. 2008. Benchmarking textual annotation tools for the semantic web. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC)*.

M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. 2002. Mnm: Ontology driven semi-automatic and automatic support for semantic markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW)*.

Towards a Psycholinguistic Database for Modern Standard Arabic

Sami Boudelaa[a], and William D. Marslen-Wilson[a]
[a]MRC Cognition and Brain Sciences Unit
sami.boudelaa@mrc-cbu.cam.ac.uk, William.marslen-wilson@mrc-cbu.cam.ac.uk

Abstract

To date, there are no Arabic databases that provide distributional information about orthographically disambiguated words and morphemes. Here we present ARALEX (Arabic Lexical database), a new tool providing type and token frequency counts for vowelled Arabic surface words, stems, bigrams and trigrams. The database also provides type and token frequency for roots and word patterns. Token frequency counts are based on an automatically annotated 40 million word corpus derived from different Arabic news papers, while the type frequency counts are based on the Hans Wehr dictionary. This database is a valuable resource for researchers across many fields. It is available for the community as a web interface and as a stand alone downloadable application on: http://www.mrc-cbu.cam.ac.uk:8081/ARALEX.online/login.jsp

1. Introduction

Arabic has been gathering a lot of attention across many fields both because of its socio-political significance and because its linguistic characteristics present a sharp contrast with Indo-European languages. Important insights have been gained through the study of Arabic phonology, syntax and morphology (Ferguson, 1959; McCarthy, 1981; Plunkett & Nakisa, 1997; Boudelaa & Marslen-Wilson, 2010). These insights are largely limited however to theoretical linguistics. Experimental research fields such as psycholinguistics, cognitive neuroscience and neurolinguistics on the other hand are lagging behind in spite of their clear benefits for language learning and language rehabilitation. One of the reasons for the scarcity of research in Arabic experimental and applied language fields is the lack of reliable databases that provide information about the distributional characteristics of

the words in the language. ARALEX, the database we describe here, promises to fill this gap by providing information about the frequency of words, morphemes, and letter combinations (i.e., bigrams and trigrams) in Modern Standard Arabic (MSA). To provide the basis for understanding the structure of ARALEX, and the choices we made to develop it, we first provide a brief description of MSA highlighting its specific features that guided the development of the ARALEX architecture

2. Basic features of MSA

MSA is a Semitic language characterized by a rich templatic morphology where effectively all content words (and most function words) are analyzable into a *root* and a *word pattern*. The root is exclusively made up of consonants and is thought to convey a broad semantic meaning which will be expressed to various degrees in the different forms featuring that particular root. By contrast, the word pattern is essentially made up of vowels although a subset of consonants can be part of it. The word pattern conveys morpho-syntactic information and defines the overall phonological structure of the word (Holes, 1995; Versteegh, 1997). Unlike stems and affixes in Indo-European languages, Arabic roots and word patterns are interleaved within each other in a non-linear manner. For example the root {xtm} with the general meaning of finishing, can be interleaved with the pattern {faEal}[1] with the morpho-syntactic meaning of active perfective, to generate the surface form [xatam] finish and with the pattern {fiEaal} to give rise to the form [xitaam] termination. Although the meaning of a given surface form is not always componential, there is a reasonable amount of consistency (McCarthy, 1981).

Stems like [xatam] and [xitaam] can be further augmented with various inflectional affixes and enclitics. For instance the complex form [waxitaamuhaa] and its termination consists of the proclitic [wa] and, the surface form or stem [xitaam] termination, the ending [u] nominative marker, and the third person feminine singular possessive pronoun enclitic [ha] her/its.

The non-concatenative nature of the Arabic morphological system makes it an interesting subject of research for experimental and applied research disciplines. It raises important questions with far reaching consequences both for cognitive and neurocognitive theories of language representation and processing. For example, are the component morphemes of an Arabic word represented independently at a cognitive and neural level? Can morphology as a domain of knowledge affect language learning? What kind of neuro-cognitive challenges are raised by the process of reading unvowelled Arabic? Addressing these kinds of issues in the context of Arabic can be promoted by a lexical database like ARALEX that makes the design of well controlled experiments easy and efficient.

3. ARALEX architecture

Any Arabic lexical resource that does not provide statistical information about roots and word patterns is bound to be of limited interest not only to psycholinguists and cognitive neuroscientists, but also to language learners and practitioners in general. For this reason we designed ARALEX to provide the typical token frequency information about surface forms, bigrams and trigrams along with information about the type and token frequencies of morphemes (i.e., roots and patterns). ARALEX relies on two sources of information: (a) the Hans Wehr Dictionary of Modern Arabic (Wehr, 1994) as used in the dictionary of stems that comes with the Arabic Morphological Analyzer (Buckwalter, 2002), and (b) a 40 million word corpus derived from various Arabic papers. The version of the stem dictionary we used consisted of 37, 494 different stems. These include native MSA words, assimilated and non-assimilated

[1] We will be using the standard Buckwalter transliteration scheme throughout this article.

foreign words and proper Arabic and foreign nouns. For each stem we manually established the corresponding root and word pattern. This resulted in the identification of 6804 roots and 2329 word patterns. This dictionary is used as a look-up table to guide a deterministic parsing algorithm applied to each stem in the corpus.

The corpus consisted of 40 million words drawn from various Arabic online newspapers . The most challenging aspect of the corpus was the absence of vowel diacritics, which makes Arabic extremely ambiguous. In order to provide accurate frequency measures of surface forms and morphemes it was necessary to disambiguate the corpus by reinstating the missing vowels. Accordingly, we first stripped the corpus off its html tags, sliced it into manageable text files, and then submitted to AraMorph (Buckwalter, 2002). For each word defined as a string of letters with space on either side of it, AraMorph outputs (a) a vowelled solution of all the possible alternatives, (b) an exhaustive parse of the word into its component morphemes, and (c) a part-of-speech tag for each solution along with an English gloss.

To choose the correct solution for each word, we developed a novel automated technique using Support Vector Machines (Wilding, 2006). The output of this technique is a probability score reflecting the accuracy of the automatic vowelling, and an entropy score which measures the amount of uncertainty in the probability score[2]. In Initial testing on 792 K words from the Arabic Treebank, the accuracy of this automatic vowelling procedure was 93% when case endings were not taken into account, and over 85% when case endings were included. When applied to the full corpus, the procedure was accurate 80% of the time for fully diacritized forms and 90% of the time accurate for forms without case endings. These figures were further cross-validated against a randomly chosen 500 K words of automatically vowelled words that were also hand-annotated by a team of native Arabic speakers in Egypt[3]. The validation showed an overall accuracy of 77.9% suggesting that the solutions chosen by the human annotators were also likely to be chosen by the automatic vowel diacritizer.

4. Combining the corpus and the dictionary

To provide type and token frequency counts, we combined the corpus and the dictionary into an integrated database[4]. For every item in the corpus which has a stem in the dictionary, we determined the root and the word pattern using the dictionary as a deterministic look-up table. Around 0.44% of the corpus stems are not listed in the dictionary, and consequently we do not provide type frequency for such items but we do provide a token frequency. For the remaining 99.56% of the data we provide frequency counts for the orthographic form, the unpointed stem (i.e., the stem without vowels)[5], the pointed stem, the root, the word pattern, the bigram and trigram frequency of the orthographic form, the root and the word pattern.

The orthographic form is defined as the graphic entity written with white space on either side of it. For instance the phrase وسيتطلب [wsytTlb] and it will require is an orthographic form. The unpointed stem is the output of AraMorph once the clitics and the affixes have been removed. In the example above the unpointed stem is [tTlb] while the pointed stem is [taTal~ab]. The root for this stem is {Tlb} and the pattern {tafaE~al}.

The token frequency statistics are computed from occurrence counts in the 40 million word corpus as the rate of occurrence per 1 million words of text, given by:

$$\text{Freq}(w) = \frac{occ(w)}{T/k}$$

where *occ(w)* is the number of occurrences of word *w* in the corpus, *T* is the total number of words in the corpus, and $k = 1,000,000$. The generation of token frequencies for orthographic forms consists simply in counting and normalizing the number of times each distinct orthographic form occurs in the corpus. Where the token. frequencies of stems, roots and word patterns are concerned, the following procedure was followed: For each record in the corpus the pointed and unpointed stems are extracted, then their corresponding root and word pattern are located in the dictionary, and the occurrence of each of these four units (i.e., the pointed stem, unpointed stem, the root, and the word pattern) is recorded. If a pointed stem is not found in the dictionary, the unpointed stem is used to match on dictionary entries without diacritics to get a set of pointed stem candidates. Then all corresponding roots and patterns for that set of stems are located and recorded, thus increasing recall at the cost of potentially decreasing precision.

The type frequencies of roots and patterns are simply raw counts and are extracted from the dictionary. Finally the character n-gram frequencies (bigrams and trigrams) are computed from the 40 million word corpus for orthographic forms, root, and word patterns as follows:

$$\text{Freq}(g) = \frac{occ(g)}{T/k}$$

where *occ(g)* is the number of occurrences of n-gram *g* in the corpus, *T* is the total number of n-grams in the corpus, and $k = 1,000,000$. .

5. Aralex interface

Two interfaces are developed to support the use of ARALEX: A JSP/Java-based web interface, and a Java-Based Command-Line Interface (CLI). Both are based on the Apache Lucene index tool (http://lucene.apache.org/java) and provide advanced query functionality with rapid response times.

The Web interface is aimed at the majority of users whose needs can be met by a set of predefined queries. It allows the user to query the database using either Buckwalter's transliteration scheme or Arabic script. Users can request the surface frequency for an orthographic form, a pointed stem, an unpointed stem, a root and a word pattern. They can also request the type frequency for roots and patterns, and the bigram and trigram frequencies for orthographic forms, roots and patterns. A list of items with specific characteristics can also be obtained. The output can be sorted by a search unit (e.g., orthographic form frequency or root type frequency) ordered in ascending or descending order. All the user needs to do is to enter a search term in the appropriate window, and tick the appropriate boxes or indeed check all the boxes to have exhaustive information about the search string, and hit the search button.

The CLI offers a powerful, customizable method for querying ARALEX. The input to CLI can be a single word or a text file,

[2] For more details the reader is referred to Wilding, 2006.

[3] The hand annotation was conducted under the leadership of Dr Sameh Al-Ansary of Alexandria University, Egypt.

[4] The integration of the corpus and the dictionary, and the development of the front-end interface for ARALEX were done in collaboration with Ted Briscoe and Ben Medlock, in a contract with the iLexIR company.

[5] We use the terms "unpointed", "unvowelled" and "non-diacritized" interchangeably.

allowing batch processing, and the output can be written into a file or displayed on the screen. To use the ARALEX CLI, the user needs to install Java JDK 5.0 or later, and Lucene 2.3.2 or later. An ARALEX command-line interface with Java class files and an ARALEX Lucene database index are also required and can be downloaded from the ARALEX website. Once these components are available and the Lucene core JAR is on the system classpath for the ARALEX CLI, the interface can be invoked by the command *java SearchDB*. If successful, this should display the input argument format, options, and field names. At this stage the program requires the directory containing the ARALEX index files to be specified. Invoking the command *java SearchDB index_dir*, where *index_dir* is the location of the database index, yields the prompt *Enter query*. From now on, any valid Lucene query can be entered (for further details refer to Boudelaa & Marslen-Wilson, 2010).

6. Conclusion

ARALEX is the first Arabic lexical database to provide frequency information about vowelled words, morphemes and letter and phoneme bigrams. It allows experimental researchers to design well controlled experiments, and provides a valuable source of information for natural language processing development. It can also be used to derive basic and/or more advanced vocabulary lists tailored to the needs of various language learners.

7. Acknowledgements

This work is supported by a British Academy Large Research Grant (LRG42466) and MRC (grants U.1055.04.002.00001.01). The authors would like to thank Sameh Al-Ansary, Ted Briscoe, Tim Buckwalter, Hubert Jin, Mohamed Maamouri, Fermin Moscoso del Prado Martin, Dilworth B. Parkinson and Mark Wilding for their help at different stages of the project.

8. References

Boudelaa, S., & Marslen-Wilson, W.D. (2010). ARALEX: A lexical database for Modern Standard Arabic. *Behavioral Research Methods in press*

Boudelaa, S., Pulvermüller, F., Hauk, O., Shtyrov, Y. & Marslen-Wilson, W.D. (2010). Arabic morphology in the neural language system. *Journal of Cognitive Neuroscience in press*

Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. *Linguistic Data Consortium, catalog number LDC2002L49*, ISBN 1-58563-257-0.

Ferguson, C. A. (1959). Diglossia. *Word, 15*, 325-340.

Holes, C. (1995). *Modern Arabic: Structures, functions and varieties*. London and New York: Longman.

Maamouri, M., & Bies, A. (2004). Developing an Arabic Treebank: methods, guidelines, procedures, and tools. Proceedings of the Workshop on Computational Approaches to Arabic Script-based languages, p. 2-9. Geneva, Switzerland.

McCarthy, J. J. (1981). A prosodic theory of non-concatenative morphology. *Linguistic Inquiry, 12*, 373-418.

Plunkett, K. & Nakisa, R.C. (1997). A connectionist model of the Arabic plural system. *Language and Cognitive Processes, 12*, 807-836.

Versteegh, K. (1997). *The Arabic Language*. Edingburgh University Press.

Wehr, H. (1994). *Arabic-English Dictionary*. Spoken Language Services, Inc., Ithaca, NY.

Wilding, M. (2006). *Bootstrapping Arabic pointing and morphological structure.* MPhil in Computer Speech, Language and Internet Technology. University of Cambridge, St Catherine's College.

Creating Arabic-English Parallel Word-Aligned Treebank Corpora at LDC

Stephen Grimes, Xuansong Li, Ann Bies, Seth Kulick, Xiaoyi Ma, Stephanie Strassel
Linguistic Data Consortium, University of Pennsylvania
3600 Market Street, Suite 810, Philadelphia, PA USA
E-mail: { sgrimes, xuansong, bies, skulick, xma, strassel}@ldc.upenn.edu

Abstract

This contribution describes an Arabic-English parallel word aligned treebank corpus from the Linguistic Data Consortium that is currently under production. Herein we primarily focus on efforts required to assemble the package and instructions for using it. It was crucial that word alignment be performed on tokens produced during treebanking to ensure cohesion and greater utility of the corpus. Word alignment guidelines were enriched to allow for alignment of treebank tokens; in some cases more detailed word alignments are now possible. We also discuss future annotation enhancements for Arabic-English word alignment.

1. Introduction

1.1 Parallel Treebanks

Multiple annotation of corpora is common in the development of computational linguistic language resources. Additional annotation increases potential information extraction from a given resource. For example, many existing parallel corpora have been developed into parallel treebanks, and for several language pairs there exist parallel treebank corpora. Parallel treebank corpora are parallel texts for which there exist manual parses for both languages (and possibly POS tags also). Examples include Czech-English (Hajic et al., 2001), English-German (Cyrus et al., 2003), English-Swedish (Ahrenburg, 2007), Swedish-Turkish (Megyesi et al., 2008), Arabic-English (Maamouri et al., 2005; Bies, 2006), Chinese-English (Palmer et al., 2005; Bies et al., 2007). The latter corpora produced by LDC are of particular note due to their high data volume.

Parallel word-aligned treebank corpora appear to be rare, and their scarcity is likely due to their being very resource-intensive to create. The most prominent related corpus is called SMULTRON and is a parallel aligned treebank corpus for one-thousand English, Swedish, and German sentences (Gustafson-Capkova et al., 2007). In SMULTRON, alignment is pairwise between each of the component languages, and annotation permitted between syntactic categories and not exclusively between words.

1.2 Current Project

The present paper discusses key points in creating an Arabic-English parallel word-aligned treebank corpus. We have also included a brief description of this corpus in the LREC 2010 Language Resource Map.

As shown in Table 1, releases for this corpus began in 2009, and to date more than 325,000 words of Arabic and the corresponding English translation have been treebanked and word aligned. Each release includes data from one or more genre: newswire (NW), broadcast news transcripts (BN), or online web resources such as blogs (WB).

1.3 Organization of the Paper

The paper is structured as follows. Section 2 discusses development of Arabic and English treebanks. Section 3 discusses word alignment at LDC. Section 4 addresses issues faced in combining treebank and word alignment annotation. Section 5 has information about the corpus structure and how to use the data. Section 6 provides a critical analysis and discussion of future directions.

Release date	Genre	Words	Tokens	Sentences
4/9/2009	NW	9191	13145	382
9/21/2009	NW	182351	267520	7711
9/21/2009	BN	89213	115826	4824
10/24/2009	NW	16207	22544	611
10/24/2009	WB	6656	9478	288
1/29/2010	BN	9930	12629	705
1/29/2010	WB	12640	18660	565
Total		326188	459802	15086

Table 1. Annotation volume as of May 2010. Figures reported for words and tokens refer to the Arabic source.

2. Development of Parallel Treebanks

The path towards construction of the resource under discussion could be considered to begin with the Arabic Treebank (ATB) corpus (Maamouri et al., 2005). Translation of the Arabic to English created parallel texts, and when the English-Arabic Translation Treebank (EATB) (Bies, 2006) is used in conjunction with the ATB, this serves as an English-Arabic parallel treebank corpus. Please refer to documentation released with these corpora for additional discussion concerning construction, annotation guidelines, and quality control efforts that went into creating the individual treebanks.

In developing parallel treebanks, care must be taken to ensure sentence segments remain parallel from the original parallel corpus. Arabic sentences are often translated as multiple English sentences. Hence one Arabic tree may correspond to multiple English trees, and occasionally effort is required to enforce that sentence segments remain parallel. For a similar project involving an English-Chinese parallel word-aligned treebanked corpus, English and Chinese treebanking were performed independently at different locations, and the resulting corpora were only weakly parallel; an automatic sentence aligner was required to re-establish the parallel texts. We used Champollion, a lexicon-based sentence aligner for robust alignment of the noisy data (Ma, 2006). Such a tool may be necessary for others creating parallel aligned

treebank corpora if the data inputs are not already sentence-wise parallel.

The Arabic Treebank (ATB) distinguishes between source and treebank tokens. While source tokens are generally whitespace-delimited words, the treebank tokens are produced using a combination of SAMA (Maamouri et al., 2009) for morphological analysis, selection from amongst alternative morphological analyses, and finally splitting of the source token into one or more treebank tokens based on clitic or pronoun boundaries.

For release as part of this corpus, the ATB and EATB are provided in Penn Treebank format (Bies et al., 1995). The trees are unmodified from ATB/EATB releases except that the tokens were replaced with token IDs. This structure is discussed in greater detail in Section 5.

3. Word Alignment Annotation

At the LDC, word alignment is a manual annotation process that creates a mapping between words or tokens in parallel texts. While automatic or semi-automatic methods exist for producing alignments, we avoid these methods. Manual alignment serves as a gold standard for training automatic word alignment algorithms and for use in machine translation (c.f. Melamed 2001, Véronis and Langlais 2000), and it is desirable that annotator decisions during manual alignment not be biased through use of partially pre-aligned tokens. It is felt that annotators may accept the automatic alignment and also lower annotator agreement at the same time.

Using higher-quality manual alignment data for training data results in better machine translations. Fossum, Knight, and Abney (2008) showed that using Arabic and English parsers or statistical word alignment tools such as GIZA++ instead of gold standard annotations contributes to degradations in training data quality that significantly impact BLEU scores for machine translation. While automatic parsing and word aligning have their place in NLP toolkits, use of manually-annotated training data is always preferred if annotator resources are available.

3.1 Word Alignment Annotation Guidelines

LDC's word alignment guidelines are adapted from previous task specifications including those used in the BLINKER project (Melamed 1998a, 1998b). Single or multiple tokens (words, punctuation, clitics, etc.) may be aligned to a token in the opposite language, or a given token may be marked as not translated. Early LDC Arabic-English word alignment releases as part of the DARPA GALE program were generally based on whitespace tokenization.

Word alignment guidelines serve to increase annotator agreement, but different word alignment projects may have unique guidelines according to what is deemed translation equivalence. For example, are pronouns permitted to be aligned to proper nouns with which they are coindexed? Our point here is to encourage the corpus user to explore alignment guidelines in detail to better understand the task.

3.3 Word Alignment and Tagging Tool

Word alignment is performed on unvocalized tokens rendered in Arabic script. LDC's word alignment tool allows annotators to simultaneously align tokens and tag them with meta data or semantic labels. A screenshot of the tool is shown in Figure 1.

The navigation panel on the right side of the software displays original (untokenized) source text to help annotators understand the context of surrounding sentences (which aids in, for example, anaphora resolution). Having untokenized source text also aids in resolving interpretation ambiguities that would arise if annotators could only see tokenized, unvocalized script.

3.3 Additional Tagging for Word Alignment

In addition to part-of-speech tags produced as part of treebank annotation, word alignment annotators have the option of adding certain language-specific tags to aid in disambiguation. A tagging task for Arabic-English has recently been added to the duties of word alignment annotators, and it is described as follows.

For unaligned words or phrases having locally-related constituents to which to attach, they are tagged as "GLU" (i.e., "glue"). This indicates local word relations among dependency constituents. The following are some cases in which the GLU tag would be used:

-English subject pronouns omitted in Arabic.

-Unmatched verb "to be" for Arabic equational sentences.

-Unmatched pronouns and relative nouns when linked to their referents.

-Unmatched possessives ('s and ') when linked to their possessor.

-When a preposition in one language has no counterpart, the extra preposition attached to the object is marked GLU.

-Two or more prepositions in one language while there is one preposition in the other side; the unmatched preposition would be tagged as GLU.

It is hoped that the presence of the GLU tag provides a clue in understanding morphology better, and we will continue to explore using additional tags for this task.

4. Uniting Treebank and Word Alignment Annotation

This section describes efforts to join treebank and word alignment annotation.

4.1 Order of Annotation

The order of annotation in creating a parallel word-aligned treebank corpus is important. From the parallel corpus, the sentences can first be treebanked or word aligned. If word alignment was to proceed first, the tokens used for word alignment would serve as input to treebanking. However, treebank tokenization includes morphosyntactic analysis, and hence treebank tokenization is only determined manually during treebank annotation. For this reason, the preferred workflow is to only perform word-alignment annotation after experienced treebank annotators have fixed tokenization,

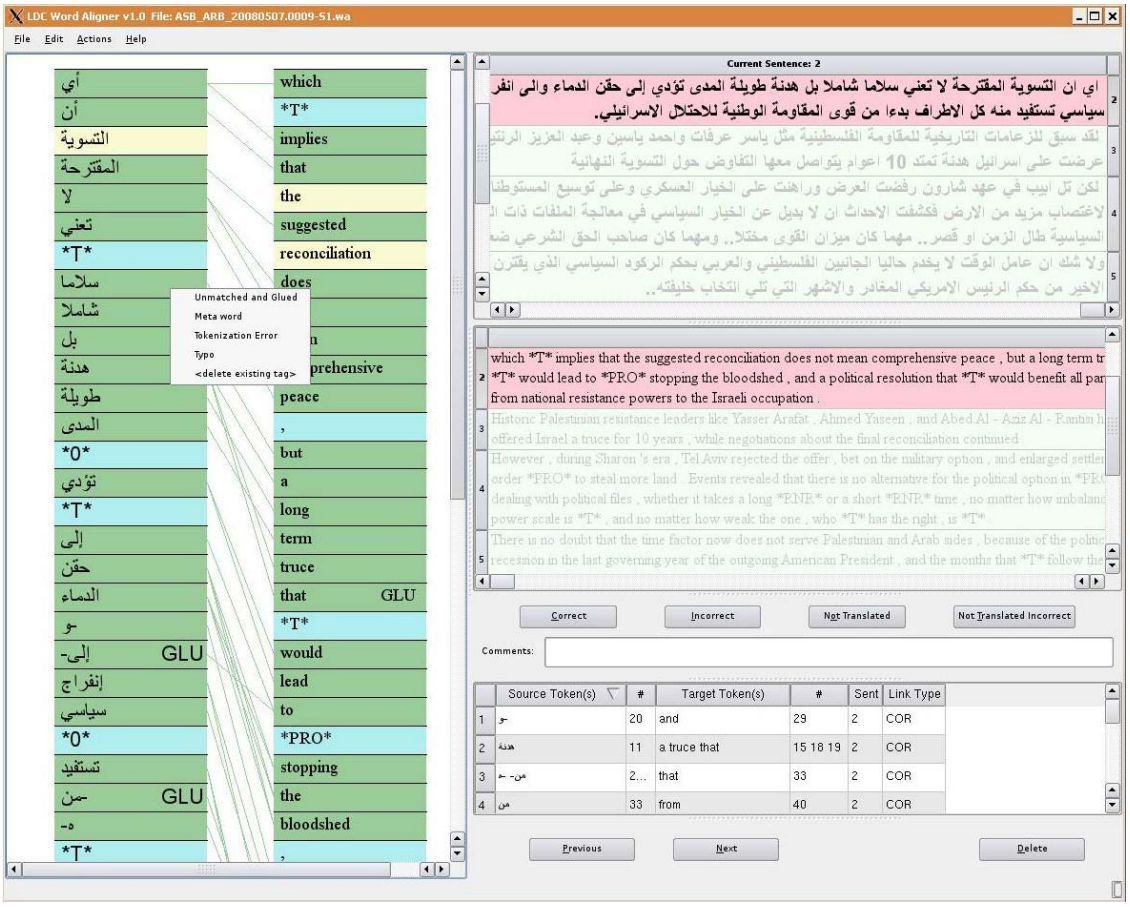

Figure 1. The PyQt-based tool used at LDC for word alignment annotation and tagging.

and it is this development trajectory we assume for the remainder of the paper.

4.2 Tokenization Modification

The word alignment guidelines were adapted so that annotation would be based on the treebank tokens instead of on source, whitespace tokens. As illustrated by the following examples, finer alignment distinctions may be made when pronouns are considered independent tokens. The example below appears in the Buckwalter transliteration[1] for convenience, but please note that the bilingual annotators work only with Arabic script.

Source:	فزجّوه بالسجن
Transliterated:	fa+ zaj~uwA +h b+ Alsijn
Morpheme gloss:	and sent.3P him to jail
Gloss:	"They sent him to jail."

Source:	سيّارته معطّلة
Transliterated:	say~Arap+h muEaT~alap
Morpheme gloss:	car his broken
Gloss:	"His car is broken."

In each case the "h" morpheme corresponding to third person singular is now considered an independent token and can be aligned to English "him" or "his" in the examples. Under previous Arabic-English word alignment guidelines , English "him" and "his" would have been aligned with the Arabic verb.

4.3 Empty Categories

In transitioning to word alignment on treebank tokens, all leaves of the syntax tree — including all Empty Categories — are considered to be tokens. This interpretation as tokens differs slightly from ATB- and EATB-defined treebank tokens which do not include the Empty Category markers such as traces, empty complementizers, and null pro markers.

Our word alignment guidelines currently dictate that all Empty Category tokens are annotated as "not translated." One could imagine amending guidelines to allow for the alignment of Empty Category markers to pronouns in the translation. This is not currently being practiced. The primary reason for including Empty Categories as tokens for word alignment is to ensure that, for each language, the number of tree leaves is identical to the number of word alignment tokens. This requirement simplifies somewhat the data validation process.

4.4 Data Validation

Validation of the data structures have both manual and automatic components.

4.4.1 Treebank validation

Throughout the Treebank pipelines, there are numerous stages and methods of sanity checks and content validation, to assure that annotations are coherent, correctly formatted, and consistent within and across annotation files, and to confirm that the resulting annotated text remains fully concordant with the original transcripts (for Arabic) or translations (for English), so that cross-referential integrity with the original data and with English translations is maintained.

For both Arabic Treebank and English Treebank, quality control passes are performed to check for and correct errors of annotation in the trees. The Corpus Search tool[2] is used with a set of error-search queries created at LDC to locate and index a range of known likely annotation errors involving improper patterns of tree structures, node labels, and the correspondence between part-of-speech tags and tree structure. The errors found in this way are corrected manually in the treebank annotation files.

In addition, the Arabic Treebank (ATB) closely integrates the Standard Arabic Morphological Analyzer (SAMA) into both the annotation procedure and the integrity checking procedure. The interaction between SAMA and the Treebank is evaluated throughout the workflow, so that the link between the Treebank and SAMA is as consistent as possible and explicitly notated for each token.

For details on the integration between the ATB and SAMA, along with information about the various forms of the tokens that are provided, see Kulick, Bies and Maamouri (2010). For a general overview of the ATB pipeline, see Maamouri, et al. (2010).

4.4.2 Word alignment validation

For word alignment, it is verified that all delivery files are well-formed. It is ensured that all tokens receive some type of word alignment annotation.

4.4.3 Validation of parallel word-aligned treebanks

To ensure consistency of the parallel aligned treebank, we verify that the set of tokens referenced by the treebank files coincides with the same set of tokens appearing the token and word alignment files.

5. Using the Corpus

This section provides information about the file format of the word-aligned treebanked data we are releasing. A typical release will contain seven files for each source document

-- Arabic source, collected from newswire, television broadcast, or on the web
-- English translation of Arabic source
-- Tokenized Arabic, resulting from treebank annotation
-- Tokenized English, resulting from treebank annotation
-- Treebanked Arabic
-- Treebanked English
-- Word alignment file

The parallel treebank is a standoff annotation with multiple layers of annotations with upper layer annotation referring to lower layer data (using character offsets). The diagram in Figure 2 shows the dependencies between files in the release.

[1] We use the Buckwalter transliteration. Details are available at http://www.qamus.org/transliteration.htm.

[2] CorpusSearch is freely available at http://corpussearch.sourceforge.net

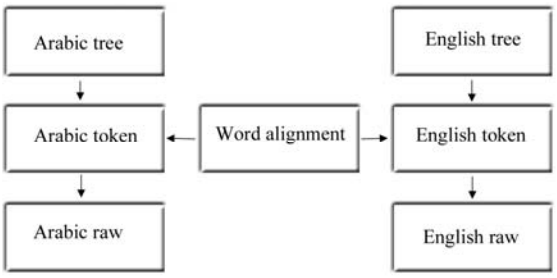

Figure 2. File structure illustration

The word alignment file and the Arabic and English tree files have token numbers which reference the Arabic and English token files. Within the token files, each token number for each sentence is expanded to give additional information. For each token in the English token files, the token number is listed, followed by a character range in the raw file to which the token corresponds, and then finally the token itself. For Arabic, multiple versions of each token are provided (unvocalized, vocalized, input string) and in multiple formats (Arabic script, Buckwalter transliteration).

We considered distributing the corpus in a single XML-based file. We felt the present structure has the following advantages:
-- the format of each type of file (raw, tokenized, tree, wa) is not modified and hence the same tools researchers wrote before can still be used;
-- the data are more easily manipulated; with XML it is necessary to fully parse the xml files for even trivial tasks;
-- it is easier and less error-prone to put the package together using separate files then using xml; and
-- separate files are more human readable.

6. Discussion

Annotator agreement for the Arabic-English word alignment task is approximately 85% after first pass annotation and higher after a quality round of annotation. In the future we plan to add additional morphosyntactic or semantic tags to the word alignment portion of the task.

We are also investigating methods for improving automatic and semi-supervised error detection. We wish to flag statistically unlikely alignments for human annotator review. Additionally, through incorporating phrase structure from treebank annotation, we might examine alignments which cross certain phrase boundaries.

7. Acknowledgements

This work was supported in part by the Defense Advanced Research Projects Agency, GALE Program Grant No. HR0011-06-1-0003. The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

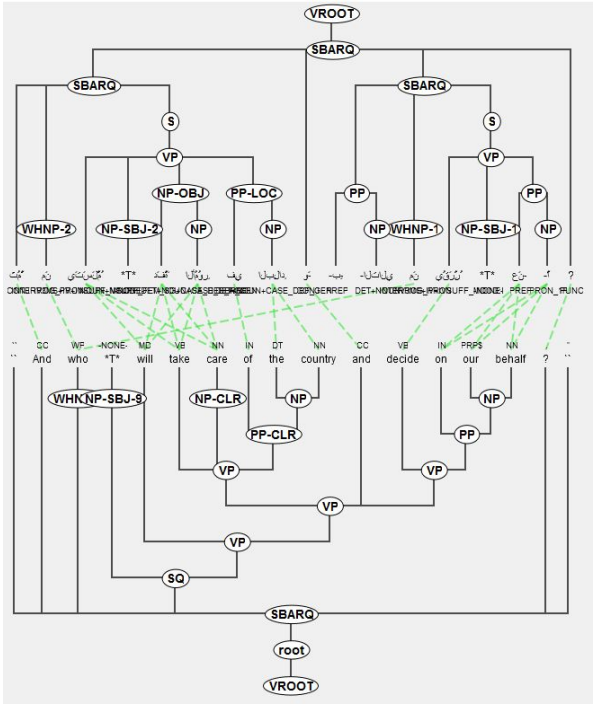

Figure 3. A view of Arabic (above) and English (below) word-aligned treebanks as displayed by TreeAligner[3].

8. References

Bies, A., Ferguson, M., Katz, K, and MacIntyre, R. (1995). Bracketing guidelines for treebank II style, Penn Treebank Project. University of Pennsylvania technical report.

Bies, A. (2006). English-Arabic Treebank v 1.0. LDC Cat. No.: LDC2006T10.

Bies, A., Palmer, M., Mott, J., and Warner, C. (2007). English Chinese Translation Treebank v 1.0. LDC Cat. No.: LDC2007T02.

Cyrus, L., Feddes, H., and Schumacher, F. (2003). Fuse – a multilayered parallel treebank. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*.

Fossum, V., Knight, K., and Abney S. (2008). Using Syntax to Improve Word Alignment Precision for Syntax-Based Machine Translation. In *Proceedings of Third Workshop on Statistical Machine Translation*, p.44-52, Columbus, June 2008. Assocation for Computational Linguistics.

Gustafson-Capkova, S., Samuelsson, Y., and Volk, M. (2007). SMULTRON (version 1.0) - The Stockholm MULtilingual parallel TReebank. Department of Linguistics, Stockholm University, Sweden.

Hajic, J., Hajicova, E., Pajas, P. , Panevova, J., Sgall, P. and Vidova Hladka, B. (2001). The Prague Dependency Treebank 1.0 CDROM. LDC Cat. No. LDC2001T10.

Kulick, S., Bies, A., and Maamouri, M. (2010). Consistent and Flexible Integration of Morphological Annotation in the Arabic Treebank. In *Proceedings of the Seventh*

[3] http://kitt.cl.uzh.ch/kitt/treealigner

International Conference on Language Resources and Evaluation (LREC 2010).

Ma, X. (2006). Champollion: A Robust Parallel Text Sentence Aligner. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation.*

Maamouri, M., Bies, A., Buckwalter, T., and Jin, H. (2005). Arabic Treebank: Part 1 v 3.0 (POS with full vocalization + syntactic analysis). LDC Cat. No.: LDC2005T02.

Maamouri, M., Bies, A., Kulick, S., Zaghouani, W., Graff, D., and Ciul, M.. (2010). From Speech to Trees: Applying Treebank Annotation to Arabic Broadcast News. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*.

Maamouri, M., Graff, D., Bouziri, B., Krouna, S., and Kulick, S. (2009). *LDC Standard Arabic Morphological Analyzer (SAMA) v. 3.0*. LDC Catalog No.: LDC2009E44. Special GALE release to be followed by a full LDC publication.

Megyesi, B., Dahlqvist, B., Pettersson, E., and Nivre, J. (2008). Swedish-Turkish Parallel Treebank. In *Proceedings of Fifth International Conference on Language Resources and Evaluation (LREC 2008)*.

Melamed, D.I. (2001). Empirical Methods for Exploiting Parallel Texts. MIT Press.

Melamed, D.I. (1998a). Annotation Style Guide for the Blinker Project. University of Pennsylvania (IRCS Technical Report #98-06).

Melamed, D.I. (1998b). Manual Annotation of Translational Equivalence: The Blinker Project. University of Pennsylvania (IRCS Technical Report #98-07).

Palmer, M., Chiou, F.-D., Xue, N., and Lee, T.-K. (2005) LDC2005T01, Chinese Treebank 5.0.

Véronis, J. and Langlais, P. (2000). Evaluation of Parallel Text Alignment Systems -- The ARCADE Project. In J. Véronis (ed.) *Parallel Text Processing, Text, Speech and Language Technology*. ordrecht, The Netherlands: Kluwer Academic Publishers, pp. 369-388.

Using English as a Pivot Language to Enhance Danish-Arabic Statistical Machine Translation

Mossab Al-Hunaity, Bente Maegaard, Dorte Hansen
Center for Language Technology
University of Copenhagen
musab@hum.ku.dk,bmaegaard@hum.ku.dk,dorteh@hum.ku.dk

Abstract

We inspect two pivot strategies for Danish-Arabic statistical machine translation (SMT) system; phrase translation pivot strategy and sentence translation pivot strategy respectively. English is used as a pivot language. We develop two SMT systems, Danish-English and English-Arabic. We use different English-Arabic and English-Danish data resources. Our final results show that SMT systems developed under sentence based pivot strategy outperforms system developed under phrase based pivot strategy, especially when common parallel corpora are not available.

1. Introduction

Developing a statistical machine translation (SMT) system between any two languages usually requires a common parallel corpus. This is used in training the SMT in translating the source language to the target language. Bilingual corpora are usually available for widely spread language pairs like Arabic English Chinese, etc, but when trying to develop SMT systems for languages pair like Arabic-Danish a bilingual corpus unfortunately doesn't exist. The limited data resources make developing SMT for Arabic-Danish a real challenge. To the best of our knowledge, there has not been much direct work on SMT for the Danish-Arabic language pair. Google Translate which is a free web translation service provides the option for translation from Danish to Arabic. Google Translate web service uses gigantic monolingual texts collected by its crawling engine to build massive language models. Aligned bilingual language resources collected through web makes it easy for Google to build SMT between any language pairs. Google performs better between languages pair which has huge common data resources like the case in English and Arabic or English and Chinese. For pairs like Arabic and Danish Google translation quality is quite less than other pairs. A possible explanation for that is the lack of common parallel available resources which control the SMT learning and performance. In our work we don't consider language resources factor alone, but also we concentrate on language specific details like syntax and morphology to tune SMT learning for our Danish-Arabic baseline. We also utilize text processing tools to enhance our baseline performance. Although a parallel corpus is not available for the Danish-Arabic pair, there are lots of parallel English-Arabic and English-Danish resources available. This makes English as a pivot language between Arabic and Danish a favorable choice. Still any language can be used as a pivot language. Our experiments use two separate corpora for Danish–English and English-Arabic SMT systems. Having English as a pivot Language we apply two different pivot strategies:

- Phrase translation pivot strategy.
- Sentence translation pivot strategy.

These methods are based on techniques developed by Utiyama, Isahara (2007), but we apply these techniques with a different perspective. We use non parallel corpora as a source of training data and not corpora with common text. We develop two baselines: Danish-English system that is piped with another English-Arabic system to translate from Danish into Arabic. Each system has different training corpus from the other. Corpora yet share or intercross partially in domain. Languages nature represents another challenge for our baseline. Our System languages are from completely different families which affect experiment results greatly. Another interesting factor is the training data resources. Many previous efforts on SMT systems with pivot language were carried on parallel corpora where data was aligned on sentence level; languages either were from the same nature like European languages Koehn (2009), or they shared a parallel data for the source pivot and target. For example

Habash and Hu (2009) used English as a pivot language between Chinese and Arabic where the three languages in their system were based on the same text. Our work differs in that we train our two systems on two different unrelated sets of data. This is due to the fact of scares parallel data resources between Danish and Arabic. Many pivot strategies are suggested in previous studies like the case with Bertoldi et. al (2008) ,Utiyama, Isahara (2007) and Habash and Hu (2009). We choose to apply our experiments on two strategies; namely phrase translation and sentence translation, due to the available data resources and to hold more control on experiments conditions. We plan to inspect further techniques on Danish Arabic SMT system in future work. Our results show that using English as a pivot language is possible with partially comparable corpora and produces reasonable results. We discover that sentence translation strategy outperforms phrase translation strategy, especially when none parallel or common resources are available. We compare our experiments results with Google Translate to judge system performance. Finally we discuss future research directions we find interesting to enhance our baseline performance. In the next section we describe related work. Section 3 presents our system description. In section 4 we describe our data and present our pivot experiments details. We present our system performance results in section 5. Finally we discuss our conclusions and future work in section 6.

2. Related Work

There has been a lot of work on translation from Danish to English Koehn (2009), and from Arabic to English Sadat and Habash(2006) , Al-Onaizan and Papineni, (2006).Many efforts were spent to overcome the lack of parallel corpora with pivot methods. For example, Resnik and Smith (2003) developed a technique for mining the web to collect parallel corpora for low-density language pairs. Munteanu and Marcu (2005) extract parallel sentences from large Chinese, Arabic, and English non-parallel newspaper corpora. Statistical machine translation with pivot approach was investigated by many researchers. For example Gispert and Mario (2006) used Spanish as a bridge for their Catalan-English translation. They compared two coupling strategies: cascading of two translation systems versus training of system from parallel texts whose target part has been automatically translated from pivot to target. In their work they showed that the phrase translation strategy consistently outperformed the sentence translation strategy in their controlled experiments. Habash and Hu (2009) used English as a pivot language while translating from Arabic to Chinese. Their results showed that pivot strategy outperforms direct translation systems. Babych et al. (2007) used Russian language as a pivot from Ukrainian to English. Their comparison showed that it is possible to achieve better translation quality with pivot approach. Kumar et al. (2007) improved Arabic-English MT by using available parallel data in other languages. Their approach was to combine word alignment systems from multiple bridge languages by multiplying posterior probability matrices. This approach requires parallel data for several languages, like the United Nations or European Parliament corpus. An approach based on phrase table multiplication is discussed in Wu and Wang (2007) .Phrase table is formed for the training process. Scores of the new phrase table are computed by combining corresponding translation probabilities in the source-pivot and pivot-target phrase-tables. They also focused on phrase pivoting. They proposed a framework with two phrase tables: one extracted from a small amount of direct parallel data; and the other extracted from large amounts of indirect data with a third pivoting language. Their results were compared with many different European language as well as Chinese-Japanese translation using English as a pivoting language. Their results show that simple pivoting does not improve over direct MT. Utiyama and Isahara (2007) inspected many phrase pivoting strategies using three European languages (Spanish, French and German). Their results showed that pivoting does not work as well as direct translation. Bertoldi et. al (2008) compare between various approaches of PBSMT models with pivot languages. Their experiments were on Chinese-Spanish translation via disjoint or overlapped English as pivot language. We believe that we are the first to explore the Danish-Arabic language pair directly in MT. We also apply pivoting techniques on none parallel text corpora.

3. System Description

In our work we develop two base lines for each experiment, Danish English and English Arabic. Translation direction is from Danish to Arabic. Moses [1] package is used for training the base lines. The system partition the source sentence into phrases. Each phrase is translated into a target language phrase. We use GIZA++ Och and Ney (2003) for word alignment.

1: Moses Package http://www.statmt.org/moses/

We use Pharaoh System suite to build the phrase table and decode (Koehn, 2004). Our language models for both systems were built using the SRILM toolkit Stolcke(2002).We use a maximum phrase length of 6 to account for the increase in length of the segmented Arabic. Our distortion limit set to 6. And finally we use BLEU metric Papineni et al. (2001) to measure performance.

4. Pivot Strategy

We use the phrase-based SMT system described in the previous section to deploy our pivot methods. We inspect two pivot strategies *phrase translation* and *sentence translation*. In both strategies we use English as the pivot language. Danish and Arabic represent source and target languages. In phrase translation strategy we directly construct a Danish-Arabic phrase translation table from a Danish-English and an English-Arabic phrase-table. In sentence translation strategy we first translate a Danish sentence into *n* English sentences and translate these *n* sentences into Arabic separately. We select the highest scoring sentence from the Arabic sentences.

4.1 Sentence Translation Experiment

The sentence translation strategy uses two independently trained SMT systems: a direct Danish-English system and a direct English-Arabic system. We translate every Danish sentence *d* into *n* English sentences *e* $\{e_1, e_2, ..., e_n\}$ using a Danish-English SMT system. Then we translate each *e* sentence into Arabic sentences *a* $\{a_1, a_2,..,a_n\}$. We estimate sentence pair feature according to formula 1 below.

$$S(s,t) = \sum_{n=1}^{8}(\alpha s_n \beta s_n + \alpha t_n \beta t_n) \quad .. 1$$

$\alpha s_n \beta s_n$, $\alpha t_n \beta t_n$ is the feature functions for the source and target (s, t) sentences respectively. Feature functions represents: a trigram language model probability of the target language, two phrase translation probabilities (both directions), two lexical translation probabilities (both directions), a word penalty, a phrase penalty, and a linear reordering penalty. Further details on these feature functions is found in (Koehn, 2004; Koehn et al., 2005). We choose to limit the number of the translation for any Danish sentence to English into three due to performance issues.

1: JTextPro http://sourceforge.net/projects/jtextpro/
2: UN Corpus http://www.uncorpora.org/

We pass the translation with maximum feature score as input to the English-Arabic system.

4.2 Phrase Translation Experiment

In the phrase translation strategy we need to construct a phrase table to train the phrase-based SMT system. We need a Danish-English phrase table and an English-Arabic phrase-table. From these tables, we construct a Danish-Arabic phrase table. We use a matching algorithm that identifies parallel sentences pairs among the tables. This process is explained in Munteanu and Marcu (2005). We identify candidate sentence pairs using a word-overlap filter tool [1]. Finally we use a classifier to decide if the sentences in each pair are a good translation for each other and update our Danish-Arabic phrase table with the selected pair.

4.3 Data

Data collection was a great challenge for this experiment. Our data resources are from two groups; Arabic-English and English-Danish. Table 1 shows a brief description of our data resources. English-Arabic corpora domain intercrosses with the English-Danish corpora domain to some reasonable degree.

Name	Direction	Domain	Size (words)
Acquis	Danish-English	Legal issues / News	7.0 M
UN multilingual corpus	Arabic-English	Legal issues / News	3.2 M
Meedan	Arabic-English	News	0.5 M
LDC2004T17	Arabic-English	News	0.5 M

Table 1: Corpus resources

	Sample	Lines	Words
Training	Small	30 K	1 M
	Medium	70 K	2 M
	Large	100 K	3 M
Test	Test (Parallel)	1 K	19 K

Table2: Training and testing data sizes

For the Arabic English we selected three major resources, the United Nations (UN) multilingual corpus [2] which is available at the UN web site.

It enjoys a good quality of translation and it contains about 3.2 M lines of data and about 7 M words. The second resource was Meedan [1] corpus, which is a newly developed Arabic English corpus mainly compiled from the internet and news agencies, it contains more than 0.5 M Arabic words. The third resource was provided by LDC [2] (catalog no. LDC2004T17), it contains more than 0.5 M words, it also cover news domain. For the English Danish category we selected the Acquis [3] Corpus, it contains more than 8 K documents and more than 7 M words. Acquis contain many legal documents that cover many domains. English Arabic resources were extracted and aligned using Okapi [4] translation memory editor. With the Acquis corpus we used the available tools that are available at the Acquis website for extracting and aligning Danish English text. All data were tokenized and lowercased separately. In order to inspect the size factor on our SMT system data were compiled into three sets: Large, Medium and Small. Table 2 illustrates the training data size for each set. For testing data we collected a parallel Arabic-English-Danish text from the UN Climate Change conference 2009 which was held in Copenhagen [5]. We extracted 1 K sentences for each language. Table 2 illustrates the training data size for each experiment. The English Arabic corpora domain intercrosses with the English Danish corpora domain to some reasonable degree. We are aware that there might be some bias among data resources coverage, but due to data availability our corpora can still serve our experiments objectives. Given the expense involved in creating direct Arabic-Danish parallel text and given the large amounts of Arabic-English and English-Danish data, we think our approach in collecting data for our experiment is still valid and interesting.

5. Results and Evaluation

We measure our system performance using BLEU scores Papineni et al. (2002). We compare our system performance with Google Translate web service. Comparison with Google provides us with a general performance indicator for our system. Table 3 presents our direct translation system results for DA-EN and EN-AR baselines. As expected BLEU scores will increase when we increase the training data size.
We use the same testing data described in section 4.3 with Google Translate; results are described in Table 4. Google outperforms our direct system results especially for the EN-AR direct translation

Training Data Size	**DA-EN**	**EN-AR**
Small	20.3	25.1
Medium	21.4	26.3
Large	23.1	27.1

Table3: BLEU Scores for Direct Sentence Based SMT systems.

Our direct system for DA-EN system BLEU score was 23 which is (64%) of Google system BLEU scores while for the EN-AR system BLEU score was 27.1 which is (40%) of Google system BLEU scores.

	DA-EN	**EN-AR**	**DA-AR**	**DA-EN-AR**
Test Sample	36.0	67.0	30.0	30.0

Table 4 describe the BLEU scores for Google translate web service on our test sample

In Table 5 we present the results of the sentence pivoting system and the phrase pivoting system. Sentence based strategy outperform Phrase based strategy. For the large size training data set the system achieved a score of 19.1 for the sentence based system compared with 12.9 to the phrased based strategy .This results differs from previous similar studies like Utiyama and Isahara (2007) and Habash and Hu (2009) where pivot strategy outperform sentence strategy. Pivot system was not better because of the quality and quantity of the DA-EN-AR phrase table entries which was received from the matching algorithm. Pivot system is dependent on the matching algorithm and enhancing it will enhance system performance. Google DA-EN and DA-EN-AR results were the same. This is a good indicator that Google uses pivot approach between languages with limited resources like the case of Arabic and Danish. Figure 1 represents a sample of our best performing system results, compared with Google translate web service. The sample shows both original text and its translation, and our system translation results for the same text.

1:Meedan http://github.com/anastaw/Meedan-Memory
2: LDC http://www.ldc.upenn.edu/
3: Acquis http://langtech.jrc.it/JRC-Acquis.html
4: Okapi http://okapi.sourceforge.net/
5: Cop15 http://en.cop15.dk

Size	Sentence Based Pivot Strategy (Da- En- Ar)	Phrase Based Pivot Strategy (Da- En - Ar)
Small	15.0	11.4
Medium	16.9	12.3
Large	19.1	12.9

Table5: BLEU Scores for Phrase based and Sentence Based SMT systems.

6. Conclusion and Future work

Developing a SMT system between two language pairs that don't share many linguistic resources Like Danish and Arabic language pairs is a quite challenging task. We presented a comparison between two common pivot strategies; phrase translation and sentence translation. Our initial results show that sentence pivot strategy outperforms phrase strategy especially when common parallel corpora are not available. We compared our system results with Google translate web service to estimate relative progress and results were promising. In the future we plan to enhance our pivoting techniques. Phrase pivot strategy is still a promising technique we need to utilize with our baseline. Phrase Pivot strategy performs better when more parallel data resources are available, so we plan to collect more parallel training data for our baseline. We also plan to apply state of the art alignment technique and to use word reorder tools on our system training data. This will enhance our SMT system learning process. We also plan to train our SMT system to fit domain specific areas like weather, or climate domains. We target high quality pivot techniques that will help us outperform available commercial tools like Google Translate especially for domain specific SMT areas

Reference	**DA**	Jeg tror, at en af de store mangler ved Kyoto var, at den officielle delegation kom tilbage med en aftale, som de vidste aldrig ville blive vedtaget i senatet.
	EN	I think that a major shortcoming of Kyoto was that the official delegation came back with a treaty they knew was never going to make it through the Senate
	AR	وأعتقد أن أحد أوجه القصور الرئيسية في كيوتو هو أن الوفد الرسمي عاد مع معاهدة كانوا على علم أنها لن تمر خلال مجلس الشيوخ
System		وأعتقد أن أحد مشاكل الرئيسيةفي Kyoto كان الوفد الرسمي جاء يعود مع أنهم اعلم كان لن يتماعتماده
Google		اعتقد ان احد العيوب الرئيسية في كيوتو هو أن الوفد الرسمي عاد الى اتفاق مع أنهم يعرفون لن يتم اعتماده في مجلس الشيوخ
Reference	**DA**	Men selv om udledningen af drivhusgasser forventes at falde på grund af faldende aktivitet i industrien, tror de Boer ikke, det vil mindske presset på landene om at handle og underskrive en ny aftale.
	EN	But even though greenhouse gas emissions are expected to slow down as a result of shrinking industrial activities ,de Boer does not believe it will lessen the pressure on countries to act and sign a new treaty.
	AR	و على الرغم من الانبعاثات الغازية لبيت الدفيئة من المتوقع أن تنخفض نتيجة لانخفاض الأنشطة الصناعية ، دي بوير لا يعتقد أن ذلك سوف يقلل من الضغط على الدول للعمل والتوقيع على معاهدة جديدة
System		حتى على الرغم من انبعاثات غازات الحرارة من المتوقع تنخفض على أساس النشاط التنازلي,وستحدد الضغوط على البلاد لعمل على الاتفاقية جديدة .

Figure 1: Selected samples of system translation result

References

A. de Gispert and J. B. Mario, "Catalan-english statistical machine translation without parallel corpus: bridging through spanish," in Proc. of 5th International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, 2006.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In ICSLP.

Bogdan Babych, Anthony Hartley, and Serge Sharoff. 2007. Translating from underresourced languages: comparing direct transfer against pivot translation. In Proceedings of MT Summit XI, Copenhagen, Denmark.

Callison-Burch et al. (2006) Chris Callison-Burch, Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. InIWSLT.

Callison-Burch et al (2006)Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In Proceedings of HLT-NAACL'06. New York, NY, USA

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51.

Fatiha Sadat and Nizar Habash. 2006. Combination of Arabic preprocessing schemes for statistical machine translation. In Proceedings of Coling ACL'06. Sydney, Australia

H. Wu and H. Wang, "Pivot language approach for phrase-based statistical machine translation," in Proc.of the 45th Annual Meeting of the Association of Computational Linguistics. Prague, Czech Republic,2007, pp. 856–863.

Ibrahim Badr, Rabih Zbib, and James Glass 2008. Syntactic Phrase Reordering for English-to-Arabic Statistical Machine Translation. In Proc. of ACL/HLT.

Jakob Elming,2008, Syntactic Reordering Integrated with Phrase-based SMT , ACL proceedings.

Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In ACL.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In Proceedings of NAACL-HLT'07, Rochester, NY, USA

Munteanu and Marcu (2005)Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. Computational Linguistics,31(4):477–504.

Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, Roldano Cattoni, 2008 ,Phrase-Based Statistical Machine Translation with Pivot Languages, Proceedings of IWLST , USA.

Nizar Habash and Jun Hu.2009. Improving Arabic-Chinese Statistical Machine Translation using English as Pivot Language. Proceedings of the Fourth Workshop on Statistical Machine Translation , pages 173–181.

Philipp Koehn, Alexandra Birch and Ralf Steinberger: 462 Machine Translation Systems for Europe, in Proceedings of the 12th MT Summit, (Ottawa, Canada, 26-30 August, 2009), p. 65-72.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. Computational Linguistics,29(3):349–380.

Shankar Kumar, Franz Och, and Wolfgang Macherey.2007. Improving word alignment with bridge languages. In Proceedings of EMNLPCoNLL' 07, Prague, Czech Republic.

Yaser Al-Onaizan and Kishore Papineni. 2006 Distortion models for statistical machine translation.In Proceedings of Coling-ACL'06. Sydney, Australia.

Using a Hybrid Word Alignment Approach for Automatic Construction and Updating of Arabic to French Lexicons

Nasredine Semmar, Laib Meriama
CEA, LIST, Vision and Content Engineering Laboratory,
18 route du Panorama, Fontenay-aux-Roses, F-92265, France
nasredine.semmar@cea.fr, meriama.laib@cea.fr

Abstract

Translation lexicons are vital in machine translation and cross-language information retrieval. The high cost of lexicon development and maintenance is a major entry barrier for adding new languages pairs. The integration of automatic building of bilingual lexicons has the potential to improve not only cost-efficiency but also accuracy. Word alignment techniques are generally used to build bilingual lexicons. We present in this paper a hybrid approach to align simple and complex words (compound words and idiomatic expressions) from a parallel corpus. This approach combines linguistic and statistical methods in order to improve word alignment results. The linguistic improvements taken into account refer to the use of an existing bilingual lexicon, named entities detection and the use of grammatical tags and syntactic dependency relations between words. The word aligner has been evaluated on the MD corpus of the ARCADE II project which is composed of the same subset of sentences in Arabic and French. Arabic sentences are aligned to their French counterparts. Experimental results show that this approach achieves a significant improvement of the bilingual lexicon with simple and complex words.

1. Introduction

Translation lexicons are a vital component of several Natural Language Processing applications such as machine translation (MT) and cross-language information retrieval (CLIR). The high cost of bilingual lexicon development and maintenance is a major entry barrier for adding new languages pairs for these applications. The integration of automatic building of bilingual lexicons improves not only cost-efficiency but also accuracy. Word alignment approaches are generally used to construct bilingual lexicons (Melamed, 2001).

In this paper, we present a hybrid approach to align simple and complex words (compound words and idiomatic expressions) from parallel text corpora. This approach combines linguistic and statistical methods in order to improve word alignment results.

We present in section 2 the state of the art of aligning words from parallel text corpora. In section 3, the main steps to prepare parallel corpora for word alignment are described; we will focus, in particular, on the linguistic processing of Arabic text. We present in section 4 single and multi-word alignment approaches. We discuss in section 5 results obtained after aligning simple and complex words of a part of the ARCADE II MD (Monde Diplomatique) corpus. Section 6 concludes our study and presents our future work.

2. Related work

There are mainly three approaches for word alignment using parallel corpora:

- Statistical approaches are generally based on IBM models (Brown et al., 1993).
- Linguistic approaches for simple words and compound words alignment use bilingual lexicons and morpho-syntactic analysis on source and target sentences in order to obtain grammatical tags of words and syntactic dependency relations (Debili & Zribi, 1996; Bisson, 2001).
- A combination of the two previous approaches (Daille et al., 1994; Gaussier, 1995; Smadja et al., 1996; Blank, 2000; Barbu, 2004; Ozdowska, 2004). Gaussier (1995) approach is based on a statistical model to establish the French and English word associations. It uses the dependence properties between words and their translations. Ozdowska (2004) approach consists in matching words regards to the whole corpus, using the co-occurrence frequencies in aligned sentences. These words are used to create couples which are starting points for the propagation of matching links by using dependency relations identified by syntactic analysis in the source and target languages.

Machine translation systems based on IBM statistical models do not use any linguistic knowledge. They use parallel corpora to extract translation models and they use target monolingual corpora to learn target language model. The translation model is built by using a word alignment tool applied on a sentence-to-sentence aligned corpus. This model can be represented as a matrix of probabilities that relies target and source words. The Giza++ tool (Och, 2003) implements this kind of approach but its performance is proved only for aligning simple words. Approaches and tools for complex words alignment are at experimental stage (DeNero & Klein, 2008).

3. Pre-processing the bilingual parallel corpus

A bilingual parallel corpus is an association of two texts in two languages, which represent translations of each other. In order to use this corpus in word alignment, two

pre-processing tasks are involved on the two texts: sentence alignment and linguistic analysis.

3.1 Sentence alignment

Sentence alignment consists in mapping sentences of the source language with their translations in the target language. A number of sentence alignment approaches have been proposed (Brown et al., 1991; Gale & Church, 1991; Kay & Röscheisen, 1993).

Our approach to align the sentences of the bilingual parallel corpus combines different information sources (bilingual lexicon, sentence length and sentence position) and is based on cross-language information retrieval which consists in building a database of sentences of the target text and considering each sentence of the source text as a "query" to that database (Semmar & Fluhr, 2007). This approach uses a similarity value to evaluate whether the two sentences are translations of each other. This similarity is computed by the comparator of the cross-language search engine and consists in identifying common words between source and target sentences. This search engine is composed of a deep linguistic analysis, a statistical analysis to attribute a weight to each word of the sentence, a comparator and a reformulator to translate the words of the source sentence in the target language by using a bilingual lexicon.

In order to refine the result of alignment, we used the following three criteria:

- Number of common words between the source sentence and the target sentence (semantic similarity) must be higher than 50% of number of words of the target sentence.
- Position of the sentence to align must be in an interval of 10 compared to the position of the last aligned sentence.
- Ratio of lengths of the target sentence and the source sentence (in characters) must be higher or equal than 1.1 (A French character needs 1.1 Arabic characters): Longer sentences in Arabic tend to be translated into longer sentences in French, and shorter sentences tend to be translated into shorter sentences.

The alignment process has three steps:

- Exact match 1-1 alignment: In this step, the similarity between the source sentence and the target sentence is maximized by using the three criteria mentioned above.
- 1-2 or 2-1 alignments: The goal of this step is to attempt to merge the next unaligned sentence with the previous one already aligned. To confirm 1-2 or 2-1 alignments, we use only the first two criteria.
- Fuzzy match 1-1 alignment: This step consists in aligning two sentences with a low level of similarity. This aligner does not use the three criteria.

The parallel corpus is indexed into two databases. These two databases are composed of two sets of ordered sentences, one for each language. The sentence aligner uses a cross-language search to identify the link between the sentence in the source language and the translated sentence in the target language (Figure 1).

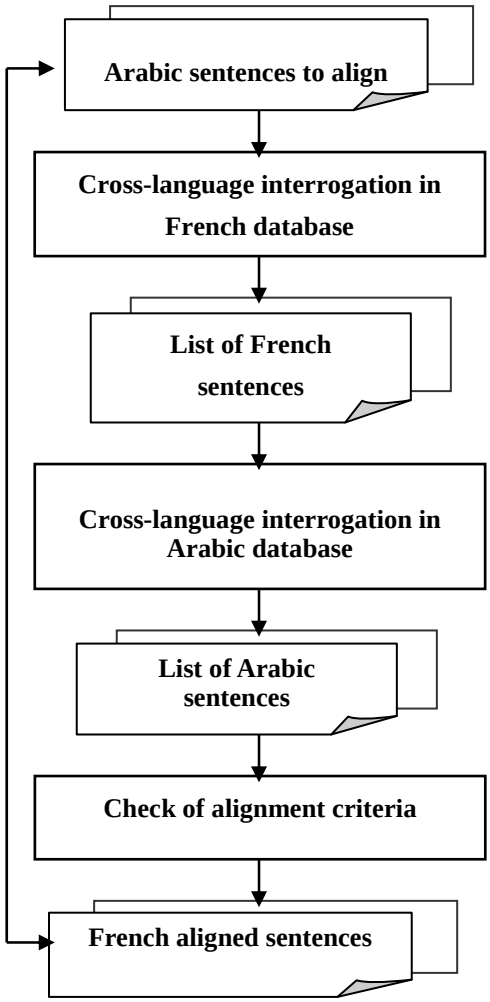

Figure 1: Sentence alignment steps.

3.2 Linguistic analysis

The linguistic analysis produces a set of normalized lemmas, a set of named entities and a set of compound words with their grammatical tags. This analysis is built using a traditional architecture involving separate processing modules:

- A morphological analyzer which looks up each word in a general full form dictionary. If these words are found, they are associated with their lemmas and all their grammatical tags. For Arabic agglutinated words which are not in the full form dictionary, a clitic stemmer was added to the morphological analyzer. The role of this stemmer is to split agglutinated words into proclitics, simple forms and enclitics.
- An idiomatic expressions recognizer which detects idiomatic expressions and considers them as single words for the rest of the processing. Idiomatic expressions are phrases or compound nouns that are listed in a specific dictionary. The detection of

idiomatic expressions is performed by applying a set of rules that are triggered on specific words and tested on left and right contexts of the trigger. These rules can recognize contiguous expressions as "البَيْت الأبْيَض" (the white house).

- A Part-Of-Speech (POS) tagger which searches valid paths through all the possible tags paths using attested trigrams and bigrams sequences. The trigram and bigram sequences are generated from a manually annotated training corpus. They are extracted from a hand-tagged corpora. If no continuous trigram full path is found, the POS tagger tries to use bigrams at the points where the trigrams were not found in the sequence. If no bigrams allow completing the path, the word is left undisambiguated. The following example shows the result of the linguistic analysis after Part-Of-Speech tagging of the Arabic sentence "في ايطاليا ادت طبيعة الاشياء الى اقناع غالبية الناخبين في طريقة غير مرئية بأن زمن الاحزاب التقليدية قد بلغ نهايته" (In Italy, the order of things has persuaded in an invisible manner a majority of voters that the time of traditional political parties was completed). Each word is represented as a string (token) with its lemma and morpho-syntactic tag (token: lemma, morpho-syntactic tag).

(1) في: فِي, Preposition
(2) ايطاليا: إِيطَالِيَا, Proper Noun
(3) ادت: أدَّ, Verb
(4) طبيعة: طَبِيعَة, Common Noun
(5) ال: ال, Definite Article
(6) اشياء: شَيْء, Common Noun
(7) الى: إِلَى, Preposition
(8) اقناع: إِقْنَاع, Common Noun
(9) غالبية: غَالِبِيَّة, Common Noun
(10) ال: ال, Definite Article
(11) ناخبين: نَاخِب, Common Noun
(12) في: فِي, Preposition
(13) طريقة: طَرِيقَة, Common Noun
(14) غير: غَيْر, Adverb
(15) مرئية: مَرْئِيّ, Adjective
(16) ب: بِ, Preposition
(17) أن: أنَ, Conjunction
(18) زمن: زَمَن, Common Noun
(19) ال: ال, Definite Article
(20) احزاب: حِزْب, Common Noun
(21) ال: ال, Definite Article
(22) تقليدية: تَقْلِيدِيّ, Adjective
(23) قد: قَد, Preposition
(24) بلغ: بَلَغَ, Verb
(25) نهايت: نَهَايَة, Common Noun
(26) ه: ه, Pronoun

- A syntactic analyzer which is used to split graph of words into nominal and verbal chains and recognize dependency relations by using a set of syntactic rules. We developed a set of dependency relations to link nouns to other nouns, a noun with a proper noun, a proper noun with a post nominal adjective and a noun with a post nominal adjective. These relations are restricted to the same nominal chain and are used to compute compound words. For example, in the nominal chain "نقل المياه" (water transportation), the syntactic analyzer considers this nominal chain as a compound word "نقل_مياه" composed of the words "نقل" (transportation) and "مياه" (water).
- A named entity recognizer which uses name triggers to identify named entities. For example, the expression "الأوَّل مِن شَهْر مَارس" (The first of March) is recognized as a date and the expression "قطر" (Qatar) is recognized as a location.
- A module to eliminate empty words which consists in identifying words that should not be used as search criteria and removing them. These empty words are identified using only their Part-Of-Speech tags (such as prepositions, articles, punctuations and some adverbs). For example, the preposition "ل" (for) in the agglutinated word "لنقل" (for transportation) is considered as an empty word.
- A module to normalize words by their lemmas. In the case the word has several lemmas, only one of these lemmas is taken as normalization. Each normalized word is associated with its morpho-syntactic tag. For example, normalization of the word "أنابيب" (pipelines) which is the plural of the word "أنْبُوب" (pipeline) is represented by the couple (أنْبُوب, Noun).

4. Word alignment

Our approach to align simple and complex words adapts and enriches the methods developed by:

- (Debili & Zribi, 1996) (Bisson, 2001) which consist to use, in one hand, a bilingual lexicon and the linguistic properties of named entities and cognates to align simple words, and on the other hand, syntactic dependency relations to align complex words.
- (Giguet & Apidianaki, 2005) which consist to use sequences of words repeated in the bilingual corpora and their occurrences to align compound words and idiomatic expressions.

4.1 Single-word alignment

Single-word alignment is composed of the following steps:

- Alignment using the existing bilingual lexicon.
- Alignment using the detection of named entities.
- Alignment using grammatical tags of words.
- Alignment using Giza++.

4.1.1. Bilingual lexicon look-up

Alignment using the existing bilingual lexicon consists in extracting for each word of the source sentence the appropriate translation in the bilingual lexicon. The result of this step is a list of lemmas of source words for which one or more translations were found in the bilingual lexicon. The Arabic to French lexicon used in this step contains 124 581 entries.

Table 1 shows results of this step for the Arabic sentence " في ايطاليا ادت طبيعة الاشياء الى اقناع غالبية الناخبين في طريقة غير مرئية بأن زمن الاحزاب التقليدية قد بلغ نهايته" and its French translation "En Italie, l'ordre des choses a persuadé de manière invisible une majorité d'électeurs que le temps des partis traditionnels était terminé".

Lemmas of words of the source sentence	Translations found in the bilingual lexicon
شَيْء	chose
غَالِبِيَّة	majorité
نَاخِب	électeur
يقَةطَر	manière
زَمَن	temps
حِزْب	parti
تَقْلِيدِيّ	traditionnel

Table 1: Single-word alignment with the existing bilingual lexicon.

4.1.2. Named entities detection

For those words that are not found in the bilingual lexicon, the single-word aligner searches named entities present in the source and target sentences. For example, for the previous Arabic sentence and its French translation, the single-word aligner detects that the Arabic word "إيطالِيَا" (Italy) and the French word "Italie" are named entities of the type "Location". However, this first step can produce alignment errors in the case the source and target sentences contain several named entities. To avoid these errors, we added a criterion related to the position of the named entity in the sentence.

4.1.3. Grammatical tags matching

If for a given word no translation is found in the bilingual lexicon and no named entities are present in the source and target sentences, the single-word aligner tries to use grammatical tags of source and target words. This is especially the case when the word to align is surrounded with some words already aligned. For example, because the grammatical tags of the words "طبيعَة" and "ordre" are the same (Noun) and "طبيعَة" is surrounded with the words "إيطالِيَ" and "شَيْء" which are already aligned in the two previous steps, the single-word aligner considers that the lemma "ordre" is the translation of the lemma "طبيعَة".

4.1.4. Giza++ alignment

For those words that are not found in the bilingual lexicon and are not aligned by named entities detection or grammatical tags matching, the single-word aligner uses results obtained with the Giza++ aligner from the bilingual parallel corpus. For example, Giza++ founds that the French word "persuasion" is a translation of the Arabic word "اقناع" despite the fact that this word does not belong to the French sentence "En Italie, l'ordre des choses a persuadé de manière invisible une majorité d'électeurs que le temps des partis traditionnels était terminé". In addition, this word has not vowels because it is taken directly from the parallel corpus. Table 2 illustrates results after running the four steps of single-word alignment.

Lemmas of words of the source sentence	Translations returned by single-word alignment
إيطالِيَا	Italie
طبيعَة	ordre
شَيْء	chose
اقناع	persuasion
غَالِبِيَّة	majorité
نَاخِب	électeur
طريقَة	manière
زَمَن	temps
حِزْب	parti
تَقْلِيدِيّ	traditionnel

Table 2: Single-word alignment results.

4.2 Multi-word alignment

The results obtained by the current tools for aligning words from parallel corpora are limited either to the extraction of bilingual simple words from specialized texts or to the extraction of bilingual noun phrases from texts related to the general field. These limitations are due to the fact that the extraction of compound words is more difficult than the extraction of simple words. The following examples illustrate some difficulties encountered when aligning compound words:

- A compound word is not automatically translated with a compound word. For example, the Arabic compound word "إعلام آلِيٌّ" is translated as a single word in French "informatique".
- The translation of a compound word is not always obtained by translating its components separately. For example, the French translation of the Arabic compound word "تَحْتَ التَّسْدِيدِ" is not "sous le règlement" but "en cours de règlement".
- A same compound word can have different forms due to the morphological, syntactic and semantic changes. These changes must be taken into account in the alignment process. For example, the Arabic compound words "إدارة موارد المياه" and " إدارة الموارد المائية" have the same French translation "gestion des resources en eau".

Our multi-word alignment approach is composed of the following steps:

- Alignment of compound words that are translated literally from one to the other.
- Alignment of idiomatic expressions and compound words that are not translated word for word.

4.2.1. Compound words alignment

Compound words alignment consists in establishing correspondences between the compound words of the

source sentence and the compound words of the target sentences. First, a syntactic analysis is applied on the source and target sentences in order to extract dependency relations between words and to recognize compound words structures. Then, reformulation rules are applied on these structures to establish correspondences between the compound words of the source sentence and the compound words of the target sentence. For example, the rule *Translation(*A.B*) = Translation(*A*).Translation(*B*)* allows to align the Arabic compound word "حزْب تَقْلِيدِيّ" with the French compound word "parti traditionnel" as follows:

Translation(حزْب.تَقْلِيدِيّ) =
Translation(حزْب).Translation(تَقْلِيدِيّ) = parti. traditionnel

In the same manner, this step aligns the compound word "طَبِيعَة_شَيْء" with the compound word "ordre_chose" even if the word "ordre" is not proposed as a translation of the word "طَبِيعَة" in the bilingual lexicon.

4.2.1. Idiomatic expressions alignment

In order to translate missed compound words and idiomatic expressions, we used a statistical approach which consists in:

- identifying the sequences of words which are candidate for the alignment: for the two texts of the bilingual corpus, we compute the sequences of repeated words and their number of occurrences.
- representing these sequences with vectors: for each sequence, we indicate numbers of segments in which the sequence appears.
- aligning the sequences: for each sequence of the source text and each sequence of the target text, we estimate the value of the translation relation with the following formula:

$$\cos(x_i, y_i) = \frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \times \sqrt{\sum y_i^2}}$$

This step results in a list of single words, compound words and idiomatic expressions of the source sentence and their translations. For example, for the previous Arabic sentence and its French translation, the multi-word aligner founds that the expression "manière invisible" is a translation of the Arabic expression "طريقة غير مرئية".

4.3 Cleaning the bilingual lexicon

The various approaches described in this paper to align simple and complex words use different tools for terminology extraction and dependency syntactic analysis. Each of these tools can be a source of noise because of errors that can be produced by the modules that compose them (POS tagging, lemmatization …). Therefore, these approaches inevitably produce incorrect matches between the words of source text and the words of target text. It thus becomes important to remove incorrect entries and retain only the correct words in the bilingual lexicons built or updated automatically by these methods.

We have established a score for each type of alignment to facilitate the cleaning process of the bilingual lexicon built or updated automatically from the parallel corpus:

- A link alignment between single words found in the bilingual corpus and validated in the bilingual dictionary has a score equal to 1.
- A link alignment between single words found by the detection of named entities (proper nouns and numerical expressions) has a score equal to 0.99.
- A link alignment between single words found by matching grammatical tags has a score equal to 0.98.
- A link alignment between single words produced by GIZA++ has a score equal to 0.97.
- A link alignment between compound words that are translated literally from one to the other has a score equal to 0.96.
- A link alignment between compound words that are not translated word for word or idiomatic expressions has a score equal to 0.95.

Table 3 presents results after running all the steps of word alignment process for simple and complex words.

Simple and complex words of the source sentence	Translations returned by word alignment	Score
إيطَالِيَا	Italie	0.99
طَبِيعَة	ordre	0.98
شَيْء	chose	1
اقناع	persuasion	0.97
غَالِبِيَّة	majorité	1
نَاخِب	électeur	1
طَرِيقَة	manière	1
زَمَن	temps	1
حِزْب	parti	1
تَقْلِيدِيّ	traditionnel	1
غَالِبِيَّة_نَاخِب	majorité_électeur	0.96
حِزْب_يّتَقْلِيد	parti_traditionnel	0.96
زَمَن_حِزْب_تَقْلِيدِيّ	temps_parti_traditionnel	0.96
طَبِيعَة_شَيْء	ordre_chose	0.96
طريقة غير مرئية	manière invisible	0.95

Table 3: Single-word and multi-word alignment results.

5. Experimental results

The word aligner has been tested on the MD corpus of the ARCADE II project which consists of news articles from the French newspaper "Le Monde Diplomatique" (Veronis et al., 2008). The corpus contains 5 Arabic texts (244 sentences) aligned at the sentence level to 5 French texts (283 sentences). The performance of the word aligner is presented in Table 4.

Precision	Recall	F-measure
0.85	0.80	0.82

Table 4: Word alignment performance.

Analysis of the alignment results of the previous sentence (Table 3) shows, in one hand, that 10 simple words (among 14), 4 compound words and 1 idiomatic expression are correctly aligned, and on other hand, 7 simple words are aligned with the bilingual lexicon, 1 simple word is aligned with named entities detection, 1 simple word is aligned by using grammatical tag matching and 1 simple word is aligned with Giza++.

For the whole corpus, 53% of words are aligned with the bilingual lexicon, 9% are aligned with named entities detection, 15% are aligned by using grammatical tags and 4% are aligned as compound words or idiomatic expressions. Consequently, 28% of the words of the source sentence and their translations are added to the bilingual lexicon.

6. Conclusion

In this paper, we have presented a hybrid approach to word alignment combining statistical and linguistic sources of information (bilingual lexicon, named entities detection, use of grammatical tags and syntactic dependency relations, number of occurrences of word sequences). The results we obtained showed that this approach improves word alignment precision and recall, and achieves a significant enrichment of the bilingual lexicon with simple and complex words. In future work, we plan to develop strategies and techniques, in one hand, to filter word alignment results in order to clean the bilingual lexicons built or updated automatically, and on other hand, to improve the recall of the statistical approach by using the existing bilingual lexicon and the results of the morpho-syntactic analysis of the parallel corpus.

7. Acknowledgements

This research work is supported by WEBCROSSLING (ANR - Programme Technologies Logicielles - 2007) and MEDAR (Support Action FP7 – ICT – 2007 - 1) projects.

8. References

Barbu, A.M. (2004). Simple linguistic methods for improving a word alignment. In *Proceedings of the 7th International Conference on the Statistical Analysis of Textual.*

Bisson, F. (2000). U Méthodes et outils pour l'appariement de textes bilingues. Thèse de Doctorat en Informatique. Université Paris VII.

Blank, I. (2000). *Parallel Text Processing : Terminology extraction from parallel technical texts.* Dordrecht: Kluwer.

Brown, P.F., Mercier, L. (1991). Aligning Sentences in Parallel Corpora. In *Proceedings of ACL 1991*.

Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R. L. (1993). The mathematics of statistical machine translation : parameter estimation. *Computational Linguistics* 19(3).

Daille, B., Gaussier, E., Lange, J. M. (1994). Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING'94).*

Debili, F., Zribi, A. (1996). Les dépendances syntaxiques au service de l'appariement des mots. In *Proceedings of the 10ème Congrès Reconnaissance des Formes et Intelligence Artificielle.*

DeNero, J., Klein, D. (2008). The Complexity of Phrase Alignment Problems. In *Proceedings of the of ACL 2008.*

Gale, W.A., Church, K. W. (1991). A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of Association for Computational Linguistics*.

Gaussier, E., Lange, J.M. (1995). Modèles statistiques pour l'extraction de lexiques bilingues. *Traitement Automatique de la Langue* 36.

Giguet, E., Apidianaki, M. (2005). Alignement d'unités textuelles de taille variable. In *Proceedings of the 4èmes Journées de la Linguistique de Corpus.*

Kay, M., Röscheisen, M. (1993). Text translation alignment. *Computational Linguistics, Special issue on using large corpora, Volume 19, Issue 1*.

Melamed, I.D. (2001). *Empirical Methods for Exploiting Parallel Texts.* MIT Press.

Och, F.J. (2003). *GIZA++: Training of statistical translation models.* MIT Press http://www.fjoch.com/GIZA++.htm.

Ozdowska, S. (2004). Appariement bilingue de mots par propagation syntaxique à partir de corpus français/anglais alignés. In *Proceedings of the 11ème conférence TALN-RECITAL.*

Smadja, F., Mckeown, K., Hatzivassiloglou, V. (1996). Translation Collocations for Bilingual Lexicons: A Statistical Approach. *Computational Linguistics* 22(1).

Semmar, N., Fluhr, C. (2007). Arabic to French Sentence Alignment: Exploration of A Cross-language Information Retrieval Approach. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources.*

Veronis, J., Hamon, O., Ayache, C., Belmouhoub, R., Kraif, O., Laurent, D., Nguyen, T. M. H., Semmar, N., Stuck, F., Zaghouani, W. (2008). *Arcade II Action de recherche concertée sur l'alignement de documents et son évaluation.* Chapitre 2, Editions Hermès.