



Màster:

Edició:

Directors:

Any de defensa:

Col·lecció: Treballs de fi de màster

Programa oficial de postgrau

"Comunicació lingüística i mediació multilingüe"

Departament de Traducció i Ciències del Llenguatge

Index

1. <u>Introduction</u>	1
2. <u>Goals and methodology</u>	7
3. <u>Modern Standard Arabic NLP. Description and State of the Art</u>	8
3.1. Sentence Segmentation	8
3.2. Tokenization	9
3.2.1. Less than words (Finite State based Arabic word segmentation)	
3.3. Morphosyntactic Tagging	12
3.3.1. PoS Tag sets	
3.3.2. Ambiguity	
3.4. Lemmatisation	16
3.4.1. Educated Text Stemmer (ETS)	
3.4.2. Lemmatization stemming algorithm	
3.5. Diacritization	18
3.6. Base Phrase Chunking	19
4. <u>Analysis and generation tools</u>	21
4.1. Computational morphology	21
4.1.1. Tools	
4.1.1.1. BAMA	
4.1.1.2. ALMORGEANA	
4.1.1.3. ELIXIRFM	
4.1.1.4. Extra: MAGEAD	
4.1.2. Toolkits	
4.1.2.1. MADA + TOKAN	
4.1.2.2. AMIRA	

4.2. Computational syntax	24
4.2.1. The Penn Arabic Treebank	
4.2.2. The Prague Arabic Dependency Treebank	
4.2.3. The Columbia Arabic Treebank	
5. <u>Evaluation and comparison of the tools</u>	27
5.1. Comparing the toolkits: MADA+TOKAN and AMIRA	27
5.2. Comparing the forest of treebanks	28
5.3. Possible improvements.....	29
6. <u>Conclusion, future work and applications</u>	32
7. <u>References</u>	34
Annex 1: List of abbreviations	
Annex 2: Arabic Unicode: U+0600 to U+06FF	
Annex 3: The Khoja tag set	
Annex 4: The PADT tag set	
Annex 5: PATB	
Annex 6: PADT	
Annex 7: CATiB	

“... its almost too perfect algebraic-looking grammar, i.e. root and pattern morphology sometimes known as transfixation. It is so algebraic that some scholars have accused the medieval Arab grammarians of artificially contriving some artificiality around it in its classical form”

(Kay, 2009, pg. 561)

ABSTRACT

In the last few years, there has been an increase of the interest on Modern Standard Arabic. There is where computational linguistics fits in. This paper analyses the intersection between Arabic and computational linguistics, focusing on text processing and the tools developed for this purpose. The fundamental functions of Arabic computational processing are: sentence segmentation, tokenization, morphosyntactic tagging, lemmatization, diacritization and base phrase chunking. After analysing each of these tasks, a study has been carried in order to elaborate a selection of tools into two groups: computational morphology (BAMA, ALMORGEANA, ELIXIRFM, MAGEAD, MADA+TOKAN and AMIRA) and computational syntax (The Penn Arabic Treebank, The Prague Treebank, and The Columbia Arabic Treebank). Finally, the evaluation of those tools establishes the differences among them, showing their advantages and disadvantages. The conclusion of this paper opens a window to future work regarding information extraction, information retrieval, summarization, question answering or Arabic as second language.

KEYWORDS

Natural Language Processing, Modern Standard Arabic, Segmentation, Tokenization, Part-of-Speech Tagging, Lemmatization, Diacritization, Parsing, Base Phrase Chunking, Computational morphology tools, Treebanks.

1. Introduction

The Arabic language is a collection of different variants across a large geographical space among which one of them has a particular status as the formal written standard used in the media, culture and educational system. The rest of variants are identified as informal spoken dialects and they are the language of communication for daily life. According to this definition, Arabic is the most prominent Semitic language in terms of the number of speakers since they are more than 242 million people, it is the official language in 22 states and its influence is even bigger; far beyond its native speakers there are more than 1.4 billion Muslims that pray every day in Arabic regardless of their native language. Besides, there are two aspects of Arabic linguistic situation that set it apart: the high degree of difference between the standard and the dialect variations and the fact that standard Arabic is not any Arab's native language. The Arabic language is considered as a definition feature of Arabic identity (Suleiman, 1994) and there is no other language that so uniquely defines the identity of such a large amount of people with so different ethnic backgrounds.

The progress of Arabic was more than surprising: it evolved from a poetic language in the Yahiliyya (الجاهلية, pre-Islamic period) to the vehicular language of science, mathematics, chemistry, astronomy, medicine and the language of government and religion.

The First Standardization of Arabic

The early grammarians in the 8th century were the precursors of the first codification of Arabic. The motivation was their admiration for the language of the Quran, which was a model of correctness for them. That was the first time that this language was standardized with a grammar defining correct usage at all linguistic levels: phonology, morphology, syntax and lexicons. By then, Classical Arabic accomplished the conditions of modern linguists to be a standardized language. This codification motivated an intellectual movement in the Arab World;

the Arabs wanted to understand their heritage of the Roman, Greek and Indian civilizations in order to get to know better their own identity. An example of all this dedication is the proliferation of books and encyclopaedias (such as the one of Al-Khawarizmi or the Aristotelian comments of Averroes / Ibn Rusd) from the 10th to the 13th centuries.

We find in this same period another interesting fact regarding the indigenous populations of the conquered countries of the Islamic Empire, as it was called in those days. The native population of those countries outnumbered the native Arabs and they spoke a lot of different languages. These contacts had an effect in the Arabic language since each group added its own accent to the Arabic that they had to learn in order to adapt to the new govern and administration. For this reason, Arab grammars felt the need of writing specific prescriptive rules with the objective of defining the “correct” Arabic.

The Second Standardization of Arabic

After the fall of the Abbasid Dynasty (1258 when the Mongol forces conquered Baghdad and finally 1517 when the Turkish conquered Cairo), the Arabic language was no longer the language of government and that translated into its demotion. Despite that, it survived and regained its vitality in the 19th century thanks to the contact with Western Europe and the creation of Modern Standard Arabic. In a nutshell, Bonaparte led a French expedition into the occupation of Egypt from 1798 until 1801. This fact exposed the Egyptians to the culture of the Western World. Mohammed Ali Pasha, who was the wālī (والي) / governor of Egypt by then, set in motion an ambitious program to modernize Egypt. Among other initiatives, he sent young Egyptian men to France to study all modern branches of science and humanities. Mohammed Ali founded a language school to facilitate the translation of the new technical terminology into Arabic. Nevertheless, there were a large amount of ideas and concepts that did not exist in Classical Arabic until that moment. The Arabic language had to evolve in order to adapt and

assimilate these changes. It was the origin of the Modern Standard Arabic whose development has continued to the present.

What is MSA? What are the dialects?

The following question that we have to ask now would obviously be: What is this new concept of MSA?

MSA is a very common term but it is very difficult to find some kind of agreement about what it actually means. We can claim – as a general definition – that MSA is a written and oral (mostly written) variation of Arabic used in formal communication and education, it is nobody's mother tongue but it is common to all the speakers regardless their native dialect. MSA is syntactically, morphologically and phonologically based on Classical Arabic; however lexically it is much more modern.

On the other hand we can find the Arabic dialects, which are the true native language forms. Each dialect is the result of the interaction between different ancient dialect of Classical Arabic and other languages that existed in the proximities (either by neighbouring or colonization contact). The dialects are primarily oral but the latter generations are introducing the writing usage mainly through the electronic media, like the huge variety of social networks. Again, there is little agreement about how many dialects exist but one classification could be into the following groups: Egyptian Arabic (Egypt and Sudan), Levantine Arabic (Lebanon, Syria, Jordan, Palestine and Israel), Gulf Arabic (Saudi Arabia – which contains a large amount of sub-dialects – Kuwait, UAE, Bahrain, Qatar and sometimes Omani it is also included), North African Arabic (Morocco, Algeria, Tunisia, Mauritania and sometimes also Libya), Iraqi Arabic, Yemeni Arabic (with elements from Levantine and Gulf variations) and Maltese Arabic.

Apart from this, it is also common to distinguish three sub-dialects within each geographical dialect related to the social organisation: city, rural and bedouin variations.

Generally, the city dialect is considered prestigious and less marked while the Bedouin one is considered rougher.

Arabs do not think of MSA and dialect as two separate languages although they have very clear the domain of each variety. This situation is linguistically called diglossia. However, there is a large space in between that is replenished with a mix of the two variations.

Origin of Computational Arabic

The computational analysis of Arabic typically means the analysis of MSA and this implies the exclusion of archaic vocabulary, orthography and morphological features usually associated to Classical Arabic and the literature and religious domains.

The Arabic computational linguistic work began in the late 1970's and early 1980's in the USA, Europe and the Middle East.

- USA: In 1981, Weidner Communications launched the first fully automated English to Arabic machine translation system, which was implemented at the Royal palace of Sultan Qaboos in Oman. Afterwards, Omnitrans of California used it with the purpose of translating the Encyclopedia Britannica to Arabic.

- Europe: Everhard Ditters (University of Nijmegen, the Netherlands) created a system for the syntactic analysis of MSA (Ditters, 1987).

- Middle East: In 1982, Mohamed Al-Shaarikh founded the first and largest Arabic NLP commercial entity in Kuwait, the Sakhr Company. After that, the IBM Scientific Centres collaborated with Arab universities and other research centres in order to carry out diverse projects in Arabic computational linguistics, for instance NLUSA (Natural Language Understanding System for Arabic) in Kuwait University. It was the first attempt to develop a question-answering system for MSA based on LFG unification formalism.

The interest in Arabic computational linguistics hasn't finished yet, quite the opposite; it is still increasing day by day.

Challenges of MSA for researchers

Arabic computational linguists have to face the particularities of the Arabic language, which were sometimes really challenging. As follows, we can find a brief sample of those intricate features:

- *Arabic diglossia*. As mentioned before, one of the more basic characteristics of Arabic is its diglossia situation motivated by the combination of Classical Arabic, Modern Standard Arabic and the diverse dialects.
- *The Arabic script*. There are no upper case letters but Arabic letters have different shapes according to their position in the word. There is a one-to-one correspondence between the symbols of the alphabet and the phonemes in the language. In spite of this regularity, Arabic script is still hard to read because of the no-representation of vowels and diacritics. This absence creates multiple ambiguities.
- *Lack of neither capitalization nor punctuation*. The absence of both case sensitivity and strict punctuation makes the task of preprocessing the Arabic texts more difficult.
- *Complex word structure*. Arabic builds complex words by concatenating affixes and clitics (each of them represents a different part of speech) to the root through a process of agglutination. Given its high degree of inflection, the great majority of Arabic words derive from a small group of roots whose number is about six thousand.
- *Arabic as a Non-configurational language*. In a configurational language such as English, the subject and the direct object occur at two different levels in the syntactic structure of a sentence. However, this is not the case in the VSO sentences in Arabic because both are at the same level dominated by the VP.

- *Arabic as a Pro-drop language.* Arabic is one of the languages – as Spanish, Italian and Korean – that allows the optionally dropping of the subject pronoun, or what is the same, “subjectless sentences”.

Goals of Arabic Computational Linguistics

Finally, it is very important to remark that the progress of Arabic computational linguistics has advantages for both Arabic and Non-Arabic environments.

On the one hand, we find some objectives of Arabic NLP in Non-Arabic environments. For example, it facilitates Non-Arabic speakers the understanding of Arabic texts, it enables the extraction of information of interest, it helps developing tools for the processing of spoken Arabic and it improves the quality of Arabic to English or any other language machine translation systems.

On the other hand, Arabic NLP is also very useful in Arabic speaking environments because it helps transferring the knowledge and the technology to the Arab World, it serves to extend the Arabic vocabulary with new concepts (as it can be done in the other direction as well) and it makes information retrieval, extraction, summarization and translation available to the Arab speaker; among a lot of other purposes.

2. Goals and methodology: The paper

The main goal of this paper is presenting an overview of Modern Standard Arabic text processing, as we know it until today. I aim at describing the principal mechanisms that allow us to process an input text in MSA. Consequently, I have looked for different tools that perform those tasks in order to study their functionality and be able to elaborate a selection with the most representative ones. The objective that I pursue along the following pages is to introduce the reader into the compelling Arabic computational linguistics field. You may notice some references to other languages, mainly English. The explanation for this fact is that many of the available literature on this topic used English as a first step to reach afterwards Arabic. The most obvious example of this is the labelling used in most of the tag sets (§3.3.1) where English is the vehicular language in the analyses.

Regarding the structure, the sections are described as follows:

The first and the second chapters explain some of the principal points of the Arabic language in the field of computational linguistics, as well as the purposes of the present dissertation.

The third chapter analyse all the different stages of Arabic text processing from sentence segmentation to tokenization, tagging, lemmatization, diacritization and parsing/base phrase chunking.

The fourth chapter selects some of the most relevant tools that make possible the stages previously mentioned. The fifth chapter is related to the fourth because it shows the comparison and the evaluation of those tools.

The sixth chapter synthesises the main applications of Arabic NLP. Finally, this chapter sums up the whole paper, gives reasons for future works on this topic and closes it through a conclusion.

3. Modern Standard Arabic NLP. Description and State of the Art

Arabic became one of the interests of computational linguistics in the late 1970s and the advances on this field haven't stopped until today. There is still a lot of work to do but I consider that the very first step is to understand what is the Natural Language Processing, what are the main processes and how does it apply to Arabic. In the following, the reader will find the description of each of the activities that make the applications of Arabic NLP possible: Sentence segmentation, tokenization, part-of-speech tagging, lemmatization, diacritization and parsing or base phrase chunking. Nevertheless these tasks are not an end by themselves but are essential for higher applications such as Information Extraction, Machine Translation or Automatic Speech Recognition, among others.

3.1. Sentence Segmentation

The objective of segmentation is identifying and annotating the different elements at all levels of analysis, that is the reason why there are different kinds of segmentation such as the morphological one, the syntactical one or the lemmatization. These three processes take place at the same time as disambiguation (§3.3.2), an essential part when it comes to assign any tags to the elements of the text.

Almost all the computational linguistics tools for NLP require splitting a running text correctly into sentences in order to facilitate the analysis. Text segmentation is the process by which we divide the text into meaningful units: words, sentences or topics. This issue is not that trivial as we can think at first sight because some written languages have explicit word boundaries while others do not.

Languages such as Arabic, Chinese, Japanese and Korean face this segmentation process as a more challenging issue than others languages written in Latin script since they do not show neither capitalization nor specific rules of punctuation. In Arabic it is common to find some

paragraphs without a single period except the one by the end. Arabic discourse is characterised by the very extended use of coordination, subordination and logical connectives and the conjunction of sentences through coordinators like *و* *wa* (“and”) and *ف* *fa* (“and” or “so”).

The presence of capital letters and punctuation marks not only helps determining sentence boundaries but also some practical applications like Name Entity Recognition, Information Extraction or Information Retrieval. Capitalization and punctuation rules define many patterns that are very useful when locating interesting information such as street addresses, proper nouns of people/companies/countries/etc, phone numbers, etc. An expert in Arabic computational linguistic in the field of Information Extraction must have insights into the structure and syntax of the Arabic language in order to identify patterns without the presence of capital letters and punctuation marks.

3.2. Tokenization

Tokenization is the process by which we identify minimal orthographic units that can be used for morphological analysis on its own. In formal terms, an Arabic word token is composed by at least one character from the alphabet¹, the set of short vowels and diacritics², the lengthening character³ and sometimes some other Arabic characters associated with Persian language^{4 5}. If there is any occurrence of isolated short vowels or diacritics, they must be either treated as punctuation or excluded from the input for the morphological analysis.

Tokenization also includes the segmentation of the clitics from the stems. This is crucial in Arabic since prepositions; conjunctions and most of the pronouns are cliticized onto stems.

¹ UNICODE: The set of 36 characters, *hamza* through *yāʾ*, or Unicode U+0621 through U+064A

² UNICODE: U+064B through U+0652

³ UNICODE: U+0640

⁴ UNICODE: A set of four extended characters: U+067E, U+0686, U+06A4, U+06AF

⁵ The influence of other languages and ethnic groups over the Arabic writing can cause the extension of this set of characters.

There are many ways to define the boundaries of how-to-tokenize; it varies according to the particles that we want to obtain/we need for each specific goal. Habash (2010) and Habash & Sadat (2006) resolved this question through what they call tokenization schemes and tokenization techniques. While the scheme determines what is the purpose of the tokenization, technique addresses how to execute it. There is a large list of different kinds of tokenization schemes, such as Simple Tokenization (ST), Orthographic Normalization (ON), Declitization (D1/D2/D3), Decliticizing the conjunction *w+* (WA), Penn Arabic Treebank Tokenization (TB), by Morphemes (MR), by Lemmas (LEM) or following the English tokenization style (very similar as glossing). Figure 1 shows an example of this variety given by Habash (2010, p 78):

	وسينهي الرئيس جولته بزيارة الى تركيا.					
Input (ST/D0)	wsynhy	Alrjys	jwith	bzyArh	Alj	trkyA
Gloss	and will finish	the president	tour his	with visit	to	Turkey
English	The president will finish his tour with a visit to Turkey.					
Scheme						
ON_{Enr}	wsynhy	Alrjys	jwith	bzyArh	Alj	trkyA
ON_{Red}	wsynhy	Alrjys	jwith	bzyArh	Alj	trkyA
D1	w+ synhy	Alrjys	jwith	bzyArh	Alj	trkyA
D2	w+ s+ ynhy	Alrjys	jwith	b+ zyArh	Alj	trkyA
D3/S1	w+ s+ ynhy	Al+ rjys	jwlh +h	b+ zyArh	Alj	trkyA
S2	w+s+ ynhy	Al+ rjys	jwlh +h	b+ zyArh	Alj	trkyA
WA	w+ synhy	Alrjys	jwith	bzyArh	Alj	trkyA
TB	w+ s+ ynhy	Alrjys	jwlh +h	b+ zyArh	Alj	trkyA
TB_{old}	w+ synhy	Alrjys	jwlh +h	b+ zyArh	Alj	trkyA
MR	w+ s+ y+ nhy	Al+ rjys	jwl +h +h	b+ zyAr +h	Alj	trkyA
LEM	Anhj	rjys	jwlh	zyArh	Alj	trkyA
LEM+TB	w+ s+ Anhj	rjys	jwlh +h	b+ zyArh	Alj	trkyA
ENX	w+ s+ Anhj _{VBP} +S _{3MS}	Al+ rjys _{NN}	jwlh _{NN} +h	b+ zyArh _{NN}	Alj _{IN}	trkyA _{NNP}

Figure 1

And for the reverse process, detokenization, we should note that is necessary in certain contexts, for example when Arabic is the output language and it is desirable to obtain orthographically correct written Arabic. No need to say that the more complex the tokenization process, the harder is the reverse detokenization.

3.2.1. Less than words (Finite State based Arabic word segmentation)

Some applications of computational linguistics as IR, IE or MT may sometimes require the segmentation of the text into tokens smaller than words, definable as morphological units. Arabic is a highly inflected language so this process helps to obtain very useful information about stems, prefixes and suffixes. Besides there are also many other particles attached to nouns such as pronouns, prepositions or conjunctions.

There are several methods for this type of segmentation. There is one of them that is widely applied; the use of a finite state machine (FSM) based decoder. Those machines process the rules that have been established by linguists (i.e. human annotators) and create a model based on statistical learning methods. These statistical approaches are able to use unsupervised learning on corpora. The corpora at their disposal is larger than the human annotated one so they would produce more effective rules and generalizations (thanks to this bigger amount of data).

Al+bayt → 2 tokens (after segmentation) DET+NOUN The +house = <i>The house</i> البيت = (بيت) + (ال)	Bayt+I → 2 tokens (after segmentation) NOUN+POSS_PRON_1S House+my = <i>My house</i> بيتي = (ي) + (بيت)
Bi+l+bayt → 3 tokens (after segmentation) PREP+DET+NOUN By+the+house = <i>By the house</i> بالبيت = (بيت) + (ال) + (ب)	Bi+bayt+I → 3 tokens (after segmentation) PREP+NOUN+POSS_PRON_1S By+house+my = <i>By my house</i> ببيتي = (ي) + (بيت) + (ب)
Li+l+bayt → 3 tokens (after segmentation) PREP+DET+NOUN To+the+house = <i>To the house</i> لالبيت = (بيت) + (ال) + (ل)	Li+bayt+I → 3 tokens (after segmentation) PREP+NOUN+POSS_PRON_1S To+house+my = <i>To my house</i> لبيتي = (ي) + (بيت) + (ل)

Figure 2⁶

Figure 2: Determining the appropriate segmentation for an Arabic word depends upon many factors including, in some cases, ideas about the roles that various affixes play in the grammatical structure of the language. This table shows the tokenization of six different combinations of the stem بيت “bayt” (‘house’) with various prefixes and suffixes.

⁶ The list of abbreviations (Annex 1) explains what stand for what in each case.

As a final remark for this section, the segmentation of Arabic text can be very convenient if we intend to continue the analyses with the extraction of PoS tags and parsing information. Given a morphologically complex language such as Arabic, segmentation is a useful tool to establish generalizations, besides it also helps reducing the issue of data sparseness.

3.3. Morphosyntactic Tagging

We define Part-of-Speech tagging as the process by which we annotate the previously segmented words with a contextually appropriate morpho-syntactic tag indicating a part-of-speech (verb, noun, adjective...), according to the tags established by a specific tag set.

Traditional Arab grammarians classify all the words of the Arabic language in a very basic three-parts distinction that are further sub-categorised: noun, verb and particle. Nevertheless, morphologically rich languages such as Arabic can produce a very large tag sets while attempting the higher degree of accuracy. This is shown in the Buckwalter tag set (Buckwalter, 2004) based on Arabic morphemes, which can reach over 330.000 morphological tags (Habash and Rambow, 2005)⁷. Obviously, the larger sets would get more complete information but the process itself become more complex. For the sake of speed and other reasons (for instance, the intended application and its goals), some researchers prefer to use smaller, reduced sets.

Some tag sets, as the Khoja one (Khoja, 2001) (Khoja, Garside and Knowles 2001), are used to stem a word if after the initial tagging it was not found in the lexicon. Stemming is a process that produces the stem/root by removing all the affixes of the word. If we applied this to Arabic, this means removing all the prefixes, suffixes and infixes. Almost all the words in

⁷ Habash and Rambow (2005, p.573): “(...) For Arabic, this gives us about 333,000 theoretically possible completely specified morphological analyses, i.e., morphological tags, of which about 2,200 are actually used in the first 280,000 words of the Penn Arabic Treebank (ATB). In contrast, English morphological tagsets usually have about 50 tags, which cover all morphological variation. (...)”

Arabic are formed through patterns and those patterns have predictable properties and meanings. In general, the combination of affixes determines the tag of a word. This is shown in (2): the prefix is “al-” (ال), which is the definite article so the following word has to be a noun. Nevertheless, in certain occasions there is no need for more than one affix to determine the tag. In (3) we see that the prefix is a “ت” (ت) which indicates the imperfect feature and the suffix is “wn” (ون) which indicates the masculine plural feature. Those affixes reveal that the word is likely to be a second person plural masculine imperfect verb.

(2) الورد *al-ward* (“the rose”)

(3) تدرسون *t(a)dr(u)sn* (“you [plural masculine] are studying”)

3.3.1. PoS Tag sets

There is not a perfect tag set for all the applications at once, so we can find lots of tag sets for Arabic (and obviously for other languages). In the following, a variety of some of those tag sets with a briefly description and example:

- o **THE BUCKWALTER TAG SET.** It was developed by Tim Buckwalter (Buckwalter, 2004), it is a form-based tag set for tokenized and untokenized text, and it has over 70 subtags that can be combined until getting around 170 tags.

(4) الجميلة *Aljmylh* “the beautiful”:

DET+ADJ+NSUFF_FEM_SG+CASE_DEF_ACC

- o **THE REDUCED TAG SET (RTS).** Also known as Bies Tag Set or PennPOS tag set (Diab, 2007), RTS was developed by Ann Bies and Dan Bikel as an experimental tag set with the objective of reducing the tag set, nevertheless it has been widely used for other academic researchers. It has 24 tags (Diab, Hacioglu and Jurafsky, 2004) classified into subsections, which are organised in four main groups (nominals, particles, verbs, and others).

(5)⁸ جميلة *jmyla* “beautiful (fem, sg)”:

(ADJ+NSUFF_FEM_SG DT+JJ)

- o **THE KULICK TAG SET.** It was developed by Seth Kulick (Kulick, 2010) and its advantages in relation to Arabic parsing process have been proved in (Kulick et al., 2006). It contains 43 tags (extending the Bies tag set to almost the double).
- o **THE EXTENDED REDUCED TAG SET (ERTS)⁹.** This is the base tag set found in the AMIRA toolkit (we will analyse it in section §4.1.2.2.), it is a reduced version of the Buckwalter tag set and it is composed by 72 tags. In comparison to the Bies tag set, ERTS adds information about number, gender, and definiteness on nominals.
- o **THE CATIB POS TAG SET.** It was developed for the Columbia Arabic Treebank project, CATiB (Habash and Roth, 2009), and it has only six POS tags (it is the simplest one): VRB (all verb types), VRB-PASS (passive-voice verbs), NOM (nominal), PROP (proper nouns), PRT (particle), PNX (punctuation marks). This minimal version pretends to speed up human annotation maintaining all the most important information by reducing the number of tags. It exists an extended version called catibEX whose advantages for parsing have been proved (Marton, Habash and Rambow, 2010). This tag set it is easily extensive, as Habash (2010) states, to the Kulick tag set at 98.5% accuracy (Habash and Roth, 2009).
- o **THE KHOJA TAG SET.** It was developed by Shereen Khoja (Khoja, 2001) (Khoja, Garside and Knowles, 2001) as “one of the earliest almost complete

⁸ Bies mapping, available from:

<http://catalog ldc.upenn.edu/docs/LDC2010T13/atb1-v4.1-taglist-conversion-to-PennPOS-forrelease.lisp>

⁹ Full description of ERTS: (Diab, 2007)

computational tag sets for Arabic” (Habash, 2010). It is a functional tag set but it doesn’t mark construct state neither has complete coverage. 103 nouns, 57 verbs, 9 particles, 7 residuals and one punctuation mark compose the whole tag set of 177 tags in total. Each tags is the result of a concatenation of one/two letters followed by specific attributes. The tags and an example of the functionality of this tag set within a sample text can be found in the annex 3.

(6) NASgMNI = singular masculine nominative indefinite adjective

(7) VIDu3FJ = third-person dual feminine jussive imperfect verb

(8) باسمه b+Asm+h ‘in his name’ → PPr_NCSgMGI_NPrPSg3M

- o **THE PADT TAG SET.** Finally, this last tag set was developed for the Prague Arabic Dependency Treebank (Hajic’, Smrž, Buckwalter, and Jin, 2005) (Smrž and Zemánek, 2002) and we can see how it works in ElixirFM¹⁰. Each tag is created by the sum of a POS part and a Feature part. The first one is a two-character component and the second one is a seven-character string where each letter relates to a specific value of the feature. All the POS and features could be consulted in Annex 4.

3.3.2. Ambiguity

The problem of morpho-syntactic disambiguation is quite easy to define but quite hard to find a solution to: some words are ambiguous in relation with their grammatical categorisation when analyzed in isolation but they are not within their own context. It is very helpful to include a disambiguation system in the tool in order to facilitate the selection of categories for the PoS tagging.

The most common approach to disambiguate a word is the quantitative one, in which there are different methodologies such as: linguistic knowledge structured in rules, statistical

¹⁰ 72. Online interface developed by (Smrž, O. (2016), Bielický, V. (2014), Buckwalter, T. (2002))

techniques (some researchers use Hidden Markov models) or a hybrid technique combining both. The latest additions to this list of quantitative methods are taggers that use artificial intelligence techniques¹¹ or neural networks¹².

An example of an Arabic disambiguation system is the one presented by Ines Turki Khemakhem, Salma Jamoussi and Abdelmajid Ben Hamadou (Khemakhem, 2010) where they used the PoS of corresponding aligned text in French and HMM in order to resolve the ambiguities of the Arabic text.

3.4. Lemmatization

Lemmatization (Diab, Hacıoglu and Jurafsky, 2004) (Alansary, Nagi & Adly, 2008) is an advanced stemming process wherein the inflectional and variant forms of a word are segmented off in order to recover the lemma that underlies the surface word form. A complete lemmatization is reached when it involves parsing off inflectional morphology and undoing morphophonology changes (i.e., changes resulting from the combination of morphemes). Arabic is a very morphologically rich language, as it was mentioned before, which implies a true challenge for certain NLP applications such as lemmatization. This task helps generating word indexes, concordances, and dictionaries from input texts. The lemmatization process makes possible that all the forms of a particular word can be found searching only for its lemma. But, what is a lemma? It is a citation form, also called canonical representative form that exists as a dictionary entry. Sometimes we produce derivational lemmas instead of lemmas because they still contain some hints of inflections. When lemmas appear in context, it could happen that some of the letters suffer transformations at the morpheme boundaries due to morphological or morphophonological rules. It is important not to confuse the terms “lemma” and “stem”: while a

¹¹ An example from English language is (Daelemans et al., 1996)

¹² An example for English is (Brent et al., 1999) and for Portuguese (Marques and Lopes, 1996)

lemma is the more basic form of a word, the stem (or root) is the part of the word that never changes within a set of words. One word can show different stems but it will always show only one lemma.

(9) رجال *rajāl* ('men') → lemma: "رجل" *rajul* ('man') / stem: "ر - ج - ل" *l-j-r*

(10) go / went → lemma: "to go" / stem: "go" or "went"

3.4.1. Educated Text Stemmer (ETS)

Eiman Tamah Al-Shammari presented in 2009 (Al-Shammari, 2009) a new Arabic lemmatizer with a high degree of accuracy that uses syntactical knowledge to decide on stemming during the analysis. She proposed an Arabic advanced stemmer called the Educated Text Stemmer. It consists of two algorithms: the first one stems the word according to its previous stop-words and the second one stems the word by removing affixes and using pattern matching in order to compare the resulting word to a similar group with a common threshold.

However, this lemmatizer could be considered in disadvantage in comparison with other systems regarding the computational expense that it requires. Besides, (Atwan et al., 2014) indicates that for some applications such as Information Retrieval, the first algorithm does not proportionate any improvement because of "the long list of affixes and the lack of stopwords lists that can distinguish between verb and noun".

3.4.2. Lemmatization stemming algorithm

Eiman Tamah Al-Shammari also presented in 2010 a lemmatization stemming algorithm. In Figure 3 (Bsoul et al., 2014, p 330), we can see "the major steps of the Arabic lemmatization algorithm where syntactic structures

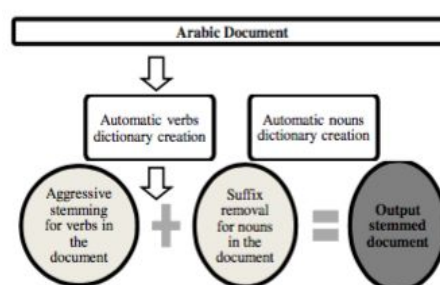


Figure 3

such as syntactic knowledge are used to determine nouns and verbs" (Bsoul et al., 2014, p 330). She also indicates the rules she established for this determining process, as we can see in Figure 4 (Bsoul et al., 2014, p 330):

RULES

- R1.** List stop words preceding verbs and nouns separately.
- R2.** All words that start with definite articles, such as "ال," are identified as nouns.
- R3.** Any word following a verb is identified as either a stop word or a noun. If this word is identified as a noun, it is added to the noun dictionary.
- R4.** Using the noun and verb corpus as a lookup table will allow identification of un-flagged nouns.

Figure 4

Seeking a better understanding of this algorithm, in Figure 5 (Bsoul et al., 2014, p 330) we have a table made by the author where it is shown how syntactic structure detects nouns and verbs. We can find the Arabic input, plus the meaning, the category of each word and the rule that helped to identify it:

Example of lemmatization algorithm					
سعره	ازداد	المنتج	طلب	ازداد	كلما
the price	It increased	the product	order	increased	whenever
noun	verb	noun	noun	verb	useful word preceding verb
based on the R4	based on the R4	based on the R2	based on the R3	based on the R1	based on the R1

Figure 5

3.5. Diacritization

Diacritization is a very basic and decisive process in Arabic. It is the task responsible of adding diacritics to the standard written form that we would find in the text. In Arabic, these diacritics are short vowels, shadda (the gemination marker), sukun (the marker of the absence of a short vowel) and tanwin (definite / indefiniteness marker at the end of a word). This process is quite close to:

- o Morphological disambiguation task (§3.3.2.) because an undiacritized word will have different morphological analyses and each of them would correspond to different diacritization schemes.
- o And also lemmatization task (§3.4.) since each different lemma will lead to different diacritization results.

An important issue to take into account in relation to diacritization is the choice of the diacritic at the last written letter of a given word form. It is an especially hard part since it requires syntactic information in order to be correctly assigned: in the case of the verbs, this last letter generally implies mood information but in the case of the nouns and adjectives it indicates the case. Often is a very hard to implement this task in a computational program, besides not all researches need it. That is the reason why there are some simpler diacritization systems that do not assign the final letter diacritic, such as DiacPart, one of the variants presented in (Roth et al, 2008). They introduce two alternative of this diacritization task implemented in their MADA system (we will see this toolkit in detail in §4.1.2.1.):

- o DiacFull: This version is the complete one, it predicts all the diacritics of a given word form. It is connected to lexeme choice and morphological tagging.
- o DiacPart: On the other hand, we find this reduced version which predicts all the diacritics of a given word form except for the final letter one. It is mainly associated with lexeme choice

3.6. Base Phrase Chunking

At last, the final task we are treating about the computational processing of Arabic is Base Phrase Chunking (Diab, 2009). This is a process with a syntactic nature whose function is creating non-recursive base phrases (like noun phrase, prepositional phrase, etc.) by concatenating a sequence of adjacent words found in the text. Base Phrase Chunking is the first step towards shallow syntactic parsing without the actual parsing itself. It is very important to specifically define the context and the target in which these base phrases are going to be used. This process is harder in languages like Arabic than in others like English due to the complexity of certain syntactic structures, such as noun phrases.

- **AMIRA and ChunkLink software**

A practical example of this task is found in the AMIRA system (we will analyse this toolkit in §4.1.2.2.). It was developed with a supervised machine learning perspective using Support Vector Machines (SVPs). The BPC component in the current version of AMIRA creates the longest possible base phrases with the minimal internal recursion. The authors have produced this technique by adapting the rules that they already had for other languages – English – to be more adequate for the Arabic language.

Here we will find 9 types of chunked base phrases that are recognised by a phrase IOB labelling (I stands for inside, O for outside and B for beginning): ADJP, ADVP, CONJP, PP, PRT, NP, SBAR, INTJ and VP. Thus, the classification is extended to a tag set of 19 tags since there are I and B labels for each chunk phrase type but only one with O label. The trained data for this system to function is obtained from the Arabic TreeBank using the ChunkLink software. ChunkLink transforms the trees into sequences of non-recursive base phrase chunks plus their corresponding IOB labels. An example that we can find is B-NP: that means that the given base phrase is a noun phrase (NP) and it is placed at the beginning of the chunk (B).

4. Analysis and generation tools

Once all the processes for Modern Standard Arabic NLP have been explained, I present in the following a selection of tools that are available for the academic, research and professional public. I established two different categories according to the morphological or syntactical character of each utility.

4.1. Computational morphology

4.1.1. Tools

4.1.1.1. BAMA

BAMA (Aliwy, 2013) (Habash, 2010), or Buckwalter Arabic Morphology analyzer, is a tool with a concatenative lexicon-driven constitution. The morphotactics and orthographic rules are directly created inside the lexicon. Thus, it won't be needed to recall both general and lexicon rules and the process would increase the speed of its performance. BAMA is built with three components: the lexicon, the compatibility tables and the analysis engine. It contains over 80.000 words, 38.600 lemmas, 3 dictionaries, and 3 compatibility tables (Elsebai, 2009, pg 75). BAMA's input is a word (with or without diacritization) that will be segmented thanks to tokenization and the output is a list of possible stems.

4.1.1.2. ALMORGEANA

ALMORGEANA (Habash and Rambow, 2005) (Habash, 2010), or ALMOR, stands for Arabic Lexeme-Based Morphological Generation and Analysis and it is a bifunctional system (it presents analysis and generation functions) elaborated on BAMA/SAMA (Standard Arabic Morphological Analyzer) databases. It analyzes to/generates from the functional level of representation (meaning lexeme and features). This fact contrasts with BAMA that focuses on the surface word form. For this reason, ALMOR increases the BAMA morphological databases by adding lexeme and feature keys and using the whole for both the analysis and the generation.

4.1.1.3. ELIXIRFM

Elixir Arabic Functional Morphology (Aliwy, 2013) (Habash 2010) is a system that presents a high-level implementation of Arabic functional morphology relying on a re-analysed version of the BAMA lexicon. Its online interface is very easy to use resulting in a very accessible tool not only for researchers but also for beginners. The main components of ElixirFM are the morphotactics, the phonology, the orthography and the tokenization levels.

4.1.1.4. Extra: MAGEAD

MAGEAD (Habash and Rambow, 2006) is an acronym of Morphological Analyzer and Generator for the Arabic Dialects. So it analyse MSA but its main target is the extension to the spoken dialects. It establishes a relation between the given word form and its lexeme and the set of linguistic features. These features are transformed into abstract morphemes, which are then ordered and finally translated into the real morphemes. MAGEAD's rules are very explicit and much less opaque than others like BAMA's or ALMORGEANA's. This has advantages and disadvantages: the con is that the system could become more complex but the pro is that the system results to be very easy to extend to any other spoken dialect. The components of MAGEAD are: lexeme and features, morphological behaviour class (MBC), morphemes and rules.

4.1.2. Toolkits

4.1.2.1. MADA+TOKAN

MADA (Habash, Rambow and Roth, 2010) (Habash, 2010) stands for Morphological Analysis and Disambiguation for Arabic. It is a tool that takes raw Arabic text as input and process it in order to add as much lexical and morphological information as possible by disambiguation, just in one step (POS tags, lexemes, diacritizations and full morphological analysis). The distinctive characteristic of MADA is that it distinguishes between morphological analysis and morphological disambiguation. MADA uses AMORGEANA internally to produce

a list of likely analyses for each word of the given text without taking into account the context. It uses up to 19 features to rank the list of potential results where each of the features gets a value in order to decide the most appropriate prediction (14 of these features use Support Vector Machine and the remaining are designed to collect information about spelling variations or n-grams statistics).

Once MADA has finished the processing of the text, TOKAN starts the tokenization and stemming tasks.

TOKAN is a general tokenizer for MSA. It takes as input the MADA disambiguated output plus a tokenization scheme according to the tokenization target. TOKAN is highly configurable so the user can decide on orthographic normalization and the order and presentation of the output depending on their posterior uses. TOKAN includes morphological back-generation through ALMORGEANA that is very useful for recreating words after the clitics have been segmented off. An example of this is ta marbuta (“ ة ”): thanks to this extension, *shayaratha* (“her car”, سيارتها) will be tokenized into *shayara* + *ha* (“car” + “her” = سيارة + ها) instead of *shayart+ha* (“car” + “her” = سيارت + ها).

4.1.2.2. AMIRA

AMIRA (Diab, 2009) (Habash, 2010) is a complete toolkit developed by Stanford University that includes: a tokenizer, a part-of-speech tagger and a base phrase chunker (or shallow syntactic parser). Its technology uses Support Vector Machines and is based on supervised learning with no explicit dependence on deep morphology knowledge. This means that it can learn generalizations from the surface data, unlike MADA. Inside AMIRA we find AMIRA-TOK and AMIRA-POS.

AMIRA-TOK is a utility whose objective is the clitic tokenization. It gets its purpose by learning generalizations obtained from the PATB (the Penn Arabic Treebank §4.2.1). The clitics that are segmented off thanks to this tool are: conjunction proclitics (*wa* / *fa* ف), prepositional

proclitics (*kaf* ك/ *al* ال/ *b* ب), future marker proclitic (*s* س), verbal partitive proclitic (*al* ال), definite article proclitic (*al* ال) and nominal enclitics indicating possessive or object pronouns. AMIRA-TOK particularity is that it faces tokenization of Arabic as a character-level chunking problem so that for each character we find an annotation composed by I (inside chunk), O (outside chunk) or B (beginning of the chunk) plus the predicted class tag (Prefix1, Prefix2, Prefix3, Suffix, Word and punctuation). AMIRA-TOK never produces non valid Arabic words.

The complementary tool in this set is AMIRA-POS. Its goal is assigning the POS tags with the ERTS (or RTS) tag set. The user can input raw or tokenized text as long as it is tokenized in a tokenize scheme consistent with one of the schemes from AMIRA-POS. Therefore, the user may ask for POS tags to be assigned to the surface forms (in this case, AMIRA-POS would have to internally run AMIRA-TOK first). Logically, the output would be presented either with or without tokenization, depending on the user's choice.

4.2. Computational syntax

4.2.1. The Penn Arabic Treebank

PATB^{13 14 15} is a project that started in 2001 at the LDC and the University of Pennsylvania, where other complex treebanks were developed such as the English one, the Chinese one or the Korean one.

All the words of the trees are classified with their corresponding full form-based morphology information. Besides, most of the sentences already had translations to English and the remaining ones have been translated and have adopted the treebank structure, which has helped to create an English-Arabic parallel treebank.

¹³ PART1 v 4.1: <https://catalog.ldc.upenn.edu/LDC2010T13>

¹⁴ PART2 v 2.0: <https://catalog.ldc.upenn.edu/LDC2004T02>

¹⁵ PART3 (full corpus) v 2.0: <https://catalog.ldc.upenn.edu/LDC2005T20>

This treebank was the first one ever built and its repercussion has been huge. It was an essential part of much research on morphological analysis, disambiguation, POS tagging, tokenization and parsing. Subsequent treebanks such as the PADT or the CATiB (which we will see in sections §4.2.2. and §4.2.3., respectively) have used it as base and inspiration for their own works. Until today, it has 3 parts that are released in the LDC catalog and 4 more under the DARPA project.

An example of this treebank can be consulted in Annex 3.

4.2.2. The Prague Arabic Dependency Treebank

PADT¹⁶ was created and is maintained by the Institute of Formal and Applied Linguistics of Charles University, in Prague. Although it used PATB as inspiration, its annotations are quite different: the POS tags are assigned according to a functional morphology tag set (the same that was developed for ElixirFM) and the syntactic annotations are in a particular dependency structure representation in two levels of information (the analytical and the tectogrammatical¹⁷).

The first version presented over 100.000 words but the current one (2.0) has reached one million of tokens-converted trees.

An example of this treebank can be consulted in Annex 4.

4.2.3. The Columbia Arabic Treebank

The last Treebank is CATiB^{18 19} and it was developed in 2008 by the Columbia University. This one is different from PADT and PATB because the priority here is to speed the

¹⁶ Institute of Formal and Applied Linguistics. 2007-2010: <http://padt-online.blogspot.com.es/>

¹⁷ “The tectogrammatical layer can be characterized as the level of linguistic (literal) meaning, i.e. as the structuring of the cognitive content proper to a particular language. On this level, the irregularities of the outer shape of sentences are absent (including synonymy and at least the prototypical cases of ambiguity) and it can thus serve as a useful interface between linguistics in the narrow sense (as the theory of language systems) on one side and such interdisciplinary domains as that of semantic interpretation (logical analysis of language, reference assignment based on inferencing using contextual and other knowledge, further metaphorical and other figurative meanings), that of discourse analysis or text linguistics, and so on, on the other.” - https://ufal.mff.cuni.cz/pdt/Corpora/PDT_1.0/Doc/tect.html

¹⁸ CATiB home webpage: <http://www1.ccls.columbia.edu/~CATiB/Home.html>

¹⁹ Downloads from CATiB website: <http://www1.ccls.columbia.edu/~CATiB/Downloads.html>

generation up with some constraints on linguistic richness. This project had two motivations: avoiding annotation of redundant linguistic information and making a more intuitive dependency structure representation by mixing some of the traditional Arabic grammar labels with the more popular ones. The result of the latter is the following set of syntactic relations: subject (SBJ), object (OBJ), predicate (PRD), topic (TPC), Idafa²⁰ (IDF), Tamyiz²¹ (TMZ), Modifier (MOD) and flat (–).

For now, CATiB has reached almost a million of tokens (270K of annotated newswire text besides the converted PATB trees from parts 1, 2 and 3) where all the sentences were taken from a parallel Arabic-English corpus (and that means that all have available their translations).

An example of this treebank can be consulted in Annex 5.

²⁰ Idafa (الإضافة): It can be referred to as *genitive* or the way Arabic expresses possession.

²¹ Tamyiz (التمييز) : The accusative of an indefinite noun which expresses a specification.

5. Evaluation and comparison of the tools

Evaluation has always been an important step in scientific research. The tools and processes shown need to be evaluated in order to assess their actual value. In this case, we will evaluate (and consequently compare) on the one hand both toolkits MADA+TOKAN and AMIRA, and on the other hand the three treebanks, which could be defined as the most important ones for Arabic computational parsing until today. The following appraisal is based on my own experience and Nizar Habash's, an expert that participated in most of the projects that have been here presented.

5.1. Comparing the toolkits MADA+TOKAN and AMIRA

In order to sum up the advantages and disadvantages of each system over the other one, the following table, Figure 6, puts in relation both systems:

	MADA + TOKAN	AMIRA
General properties		
Components	3: Almorgeana + Mada + Tokan	2: Amira-Tok + Amira POS
Steps	1) Analyze, 2) Disambiguate (tagging and tokenization at once) and 3) Generate	1) Tokenize 2) Tag
Depth of linguistic knowledge needed	It has access to deeper lexically modelled functional morphology	It is shallow in that it focuses on form-based morphology (specifically clitization) learned from annotated data
Valid Arabic words	No analysis if the word doesn't exist in Arabic	Analysis of everything
Training needs	It needs a morphological analyzer plus training data for supervised learning	It only needs annotated training data
Extensions for dialects	It will need a specific extension for the dialect	No extension needed
Specific functions		

Base Phrase Chunker	Not handled (it could be implemented in a separate module)	Handled in a separate module
Tokenization, diacritization, PoS tagging, and lemmatization	All at once in the disambiguation step	Not lemmatization nor diacritization
Degree of analysis' accuracy	Very deep	Limited
Research value		
Useful if interest in...	Exploring a large number of different sets of tokenizations and features	Limited comparisons or specific applications (they have to be compatible with AMIRA in tokenization and PoS tagging)

Finally, let's comment the performance. According to (Habash and Rambow, 2005), they show similar performance on the tasks that are shared in both systems, which are specific PATB tokenization and PoS tagging. Other than that, the speed seems to be notably faster in AMIRA. On the other hand, although MADA+TOKAN is slower, it returns a larger list of tokenizations and PoS tags than AMIRA's.

5.2. Comparing the forest of treebanks

I have built the following table (Figure 7) so the differences between the three treebanks can be more easily detected:

	PATB	CATiB	PADT
Syntactic representation	Phrases structures (PS)	Dependency structures (DS)	Dependency structures (DS)
Syntactic structure: Heads	Implicitly marked	Explicitly marked	Explicitly marked
Syntactic structure: Spans	Explicitly marked	Implicitly marked	Implicitly marked

How to express relations	Intermediate projections (e.g. VP)	Other devices (e.g. attachment labels)	Other devices (e.g. attachment labels)
Syntactic and semantic functions: labels/dashtags	20 dashtags for: <ul style="list-style-type: none"> ○ Syntactic function, e.g. –TPC or –OBJ ○ Semantic function, e.g. –TMP (time) or –LOC (location) ○ There are some which are duals, e.g. –SBJ 	Labels only for syntactic functions	Over 20 tags, different and deeper than PATB’s and CATiB’s
Empty pronouns	Annotated	Not annotated. <i>Verbs with no explicit subject can be assimilated as pro-drop (with implicit annotation)</i>	Not annotated. <i>Verbs with no explicit subject can be assimilated as pro-drop (with implicit annotation)</i>
Coreference	Annotated for traces and explicit pronouns	It does not annotate any coreference indices	Only annotates coreferences between explicit pronouns and what they corefer to
Word morphology	Over 400 PoS tags	Only 6 PoS tags	More than 400 because it is more complex and specific than PATB

In spite of all those differences that have just been exposed, it is possible the conversion between them since all the information is present in all three treebanks, the main difference is the way they are represented. Maybe the easiest to convert are PATB and PADT because they have more information at their disposal and CATiB requires a lower depth degree. The other way around would result in the lack of information, obtaining an incomplete description of the treebank.

5.3. Possible improvements

Aside from the evaluation and comparison of the tools chosen for this paper, there are some possible improvements that could be helpful in the future work.

The vast majority of the morphology tools focus on MSA or a very specific dialect, generally it the Egyptian one or some kind of gallimaufry of Levantine variations. Just one morphological tool among the ones presented, MAGEAD, addresses the issue of dialects but referring to “spoken dialect”. In the last few years, the growth of the spoken dialects into also written dialects has been quite remarkable, mainly thanks to the use of electronic media and change of mentality in the Arabic society²². I personally believe that it will be very interesting to develop a tool that implement the richness of the Arabic language, by that I mean a full covering that includes fusha or classical Arabic (the one used in the holy book, al-Qu’ran), Modern Standard Arabic and all the dialects (over at least 20 variations). Perhaps a system that has different extensions for each of the variations would be very helpful, not only because it will contain the diversity of the Arabic language, but also because the constant contact among different countries and ethnic groups is collaborating in the developing and evolution of the language. Thus, this program will have a large amount of data from different variations and it will produce analysis with a higher degree of accuracy in its predictions.

Further, in regard to treebanks and specifically word morphology, it seems unbalanced that some treebanks are huge with a high degree of specialization (PATB and PADT) while others are just superficial (CATiB). It is comprehensible that different projects have different necessities, but I think that perhaps a system that allows for various gradations within the morphology tags and also the syntactic and semantic labels. Just one system will allow representing an input in several ways, according with the needs of the user. That way, it will have access to the whole amount of data but it will only show the details relevant for the analysis and the parameters established by the user.

²² Traditionally, the dialects had less popularity since they were the language of the working-people while the language of the culture was Classical Arabic. Nowadays, dialects begin to gain defences, even more when the raise of entity-pride movements started.

Finally, I would like to point at the corpora and its necessity of update. Almost of the corpora used for the training and test of these tools are based on newspaper articles and very similar sources. In my opinion, the language is much bigger than just how journalists write (mainly if we take into account that those texts usually are very restrictive in regard to the formality and register of the language); every speaker (native or not) collaborates in the evolution of the language. This is the reason why I consider so important to compile different corpora with the real use of the language showing the great variety of the language (registers, social contexts, range of age, etc.), either written (blogs, chats, comments on the websites, social networks, etc.) or spoken (telephonic conversations, spontaneous and natural social interactions, etc.).

6. Conclusion, future work and applications

So, can we get to understand Arabic even if we don't know a word of it?

It seems a banal question to which the most logical answer would be "of course not".

However, this paper has been dedicated to prove the opposite.

We have seen in chapter 1 and 2 the general overview of Arabic, its dialects and how computational linguistics interacts with it. Arabic is a widely spread language all over the world due to several reasons (geography, culture, religion, etc.) and this fact is more than enough reason to motivate the rapprochement between Arabic and the rest of the world. Chapter 3 has introduced the stages of the Arabic text processing and what are the specific issues of each of them: sentence segmentation, tokenization, part-of-speech tagging, lemmatization, diacritization and base phrase chunking. Once we had in mind the steps to follow in order to process written Arabic input, chapter 4 has displayed a selection of tools classified into two general groups: morphological and syntactical. This description wasn't complete until chapter 5 analysed all the tools and graphically presented the differences and similarities through a table. The assessment of the tools was finished with a couple of possible improvements for the forthcoming work.

Any language is very complicated to understand if we don't get to know it first. Nevertheless, the computational linguistics field is currently under development; always with the objective of helping native Arabic to understand other languages as well as helping other speakers to understand Arabic.

In future work, it will be very stimulating and enriching to apply the processes, knowledge and tools of this paper to far-reaching applications such as Information Retrieval, Information Extraction, Summarization or Question Answering. The most exploited fields in

respect to Arabic are IR and IE (due to several necessities of the society, such as marketing or security), although they are not totally mastered yet. There are fewer resources related to summarization or question answering in Arabic, unhappily most of them are access-limited which prevents the general public from getting to know the latest advances. Regarding Question Answering, it is possible to find some studies about but it still seems to be a terrain to be explored. Another line of study totally different from the ones just mentioned is Arabic as a second language. It is not rare at all that people from across the world decide to study Arabic, either native (with a deeper knowledge of their dialect but only superficial education on the formal register, or maybe interested in other dialects different than theirs) or non-native. Nowadays, technology is an important part inside education so computational systems could positively help in this task. Some of the tools that have been here described are very useful for Arabic students (e.g. ElixirFM interface is a very intuitive tool that helps the student with the orthography and the morphology) but, for example, interactive programs could have a positive influence on the results of the learning (e.g. Champolu²³ is a website game that helps the student through a game of questions and fill-the-gap exercises).

Arabic is a captivating language with a large history in its back. Despite the enormous amount of speakers, there is still a lot of work to do. Luckily, it is at our disposal continuing the research.

²³ Champolu online learning-game: <http://champolu.com/>

7. References

1. Abouenour, L., El Hassani, S., Yazidy, T., Bouzouba, K. & Hamdani, A. 2008. 'Building an Arabic morphological analyzer as part of an open Arabic NLP platform', in *Conference: Workshop HLT & NLP within the Arabic world : Arabic Language and Local Languages Processing Status Updates and Prospects*, Language Resources and Evaluation Conference LREC, Marrakech.
2. Afli, H. 2012. 'Stanford Natural Language Processing Tools for Arabic', *Arabic Language Statistical Processing: Haithem AFLI's Blog for Volunteer Natural Language Processing scientists, especially for Arabic*. مدونة هيثم عفلي لمتطوعي الباحثين في مجال معالجة اللغات، و خاصة العربية، blog post, 19 April. Available from: <http://arabiclsp.blogspot.com.es/2012/04/stanford-natural-language-processing.html>. [18 June 2016]
3. Alansary, S., Nagi, M. & Adly, N. 2007. 'Building an international corpus of Arabic (ICA): Progress of Compilation Stage', in *7th Int. Conference on Language Engineering*, Cairo, pp 1-30.
4. Alansary, S. & Nagi, M. 2014. 'The International Corpus of Arabic: Compilation, Analysis and Evaluation', in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, Doha, pp 8-17.
5. Aliwy, A.H. 2012. 'Tokenization as Preprocessing for Arabic Tagging System', in *International Journal of Information and Education Technology*, Vol. 2, No. 4, pp 348-353.
6. Aliwy, A.H. 2013. *Arabic Morphosyntactic Raw Text Part of Speech Tagging System*, PhD thesis, Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
7. Amine, A., Bellatreche, L., Elberrichi, Z., Neuhold, E.J., Wrembel, R. (eds.). *Computer Science and Its Applications, 5th IFIP TC 5 International Conference, CIIA 2015*, Springer, Saida.
8. Alruily, M., Ayesh, A. & Zedan, H. 2013. 'Crime profiling for the Arabic language using computational linguistic techniques', in *Information Processing and Management*, pp 315-341.
9. Al-Shammari, E.T. 2009. 'Lemmatizing, Stemming, and Query Expansion Method and System', Google Patents: US 20100082333 A1
10. Al-Twairesh, N., Al-Khalifa, H. & Al-Salman, A. 2014. 'Subjectivity and sentiment analysis of Arabic: Trends and challenges', in *IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, Doha, pp 148-155. DOI: 10.1109/AICCSA.2014.7073192
11. Al-Twairesh, N., Al-Khalifa, H. & Al-Salman, A. 2015. 'Towards Analyzing Saudi Tweets', in *First International Conference on Arabic Computational Linguistics (ACLing)*, Cairo, pp 114-117. DOI: 10.1109/ACLing.2015.23
12. Arts, T., Belinkov, Y., Habash, N., Kilgarriff, A. & Suchomel, V. 2014. 'arTenTen: Arabic Corpus and Word Sketches', in *Journal of King Saud University*, Vol. 26, Computer and Information Sciences, pp 357-371.

13. Atwan, J., Mohd, M., Kanaan, G. & Bsoul, Q. 2014. 'Impact of Stemmer on Arabic Text Retrieval', in *Information Retrieval Technology. AIRS 2014, LNCS 8870*, Switzerland, pp 314 - 326.
14. Azmi, A.M. & Al-Thanyyan, S. 2012. 'A text summarizer for Arabic', in *Computer Speech and Language*, No 26, pp 260-273. DOI: 10.1016/j.csl.2012.01.002
15. Bani Khaled, T. 2014. 'Standard Arabic and Diglossia: A Problem for Language Education in the Arab World', in *American International Journal of Contemporary Research*, Vol. 4, No 8, Amman, pp 180-189.
16. Beesley, K.R., 1996. 'Arabic finite-state morphological analysis and generation', in *16th International Conference on Computational Linguistics COLING '96*, Vol. 1, Center for Sprogteknologi, Copenhagen, pp 89-94.
17. Beesley, K.R., 2001. 'Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001', in *ACL/EACL 2001*, Toulouse.
18. Belkredim, F.Z. & El Sebai, A. 2009. 'An Ontology Based Formalism for the Arabic Language Using Verbs and their Derivatives', in *Communications of the IBIMA*, Vol. 11, pp 44-52.
19. BenZeghiba, M.F., Louradour, J. & Kermorvant, C. 2015. 'Hybrid word/Part-of-Arabic-Word Language Models for arabic text document recognition', in *13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, pp 671-675. DOI: 10.1109/ICDAR.2015.7333846
20. BIES, A. & MAAMOURI, M. 2009. *Penn Arabic Treebank Guidelines*. Linguistic Data Consortium, online publication date: Jan-2009.
21. Boujelben, I., Jamoussi, S. & Ben Hamadou, A. 2014. 'A hybrid method for extracting relations between Arabic named entities', in *Journal of King Saud University-Computer and Information Sciences*, University of Sfax, Tunisia, pp 425-440.
22. Bsoul, Q, Al-Shammari, E., Mohd, M. & Atwan, J. 'Distance Measures and Stemming Impact on Arabic Document Clustering', in *Information Retrieval Technology*, Springer International Publishing, Switzerland, pp 327-339.
23. Buckwalter, T. 2004. *Buckwalter Arabic Morphological Analyzer Version 2.0*. Linguistic Data Consortium, University of Pennsylvania. LDC Catalog No.: LDC2004L02. Available from: <https://catalog.ldc.upenn.edu/LDC2004L02> [18 June 2016]
24. Champolu. Available from: <http://champolu.com/game/#> [18 June 2016]
25. Columbia University. CATiB: The Columbia Arabic Treebank. Available from: <http://www1.ccls.columbia.edu/~CATiB/Home.html> and Downloads: <http://www1.ccls.columbia.edu/~CATiB/Downloads.html> [18 June 2016]
26. Daelemans, W., Zavrel, J., Berck, P. & Gillis, S. 1996. 'MBT: A Memory-Based Part of Speech Tagger-Generator', in *Proceedings of the Fourth Workshop on Very Large Corpora*, Copenhagen, pp 14-27.
27. Diab, M., Hacioglu, K. & Jurafsky, D. 2004. 'Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks', in *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language*

- Technologies Conference (HLT-NAACL04)*, Association for Computational Linguistics, Boston, pp 149-152. DOI: 10.3115/1613984.1614022
28. Diab, M. 2007a. 'Improved Arabic Base Phrase Chunking with a New Enriched POS Tag Set', in *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, Association for Computational Linguistics, Prague, pp 89-96. DOI: 10.3115/1654576.1654592 82, 90, 112
 29. Diab, M. 2007b. 'Towards an optimal POS tag set for Moderns Standard Arabic Processing', in *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets. Cited in HABASH, N.Y. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
 30. Diab, M., Hacioglu, K. & Jurafsky, D. 2007. 'Automatic Processing of Modern Standard Arabic Text', in A. Soudi, A. van den Bosch and G. Neumann (eds.), *Arabic computational Morphology*, Springer, pp 159-179.
 31. Diab, M. (2009) 'Second Generation AMIRA Tools for Arabic Processing: Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking', Center for Computational Learning Systems, Columbia University, pp 285-288.
 32. Diab, M., Habash, N., Rambow, O. & Roth, R. 2011. 'CADIM Arabic Tools: Morphological Analysis, Disambiguation and Generation, Tokenization, Diacritization, Lemmatization, POS Tagging and Base Phrase Chunking', in Olive, Joseph, Christianson, Caitlin, McCary, John (eds.) *Handbook of Natural Language Processing and Machine Translation*, DARPA Global Autonomous Language Exploitation, Springer-Verlag New York.
 33. Elsebai, A. 2009. *A Rules Based System for Named Entity Recognition in Modern Standard Arabic*, PhD thesis, School of Computing, Science and Engineering, University of Salford.
 34. FARGHALY, A. 2010. *Arabic Computational Linguistics*. CSLI Studies in Computational Linguistics. CSLI Publications.
 35. Gelbukh, A. (ed.) 2004. *Computational Linguistics and Intelligent Text Processing. 5th International Conference, CICLing 2004*, Springer, Seoul.
 36. Green, S. & Manning, C.D. 2010. 'Better Arabic Parsing: Baselines, Evaluations, and Analysis', Computer Science Department, Stanford University.
 37. Guidère, M. 2002. 'Toward Corpus-Based Machine Translation for Standard Arabic', in *Translation Journal*, Vol. 6, No 1,
 38. Habash, N. & Rambow. 2005. 'Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop', in *Proceedings of the 43rd Annual Meeting of the ACL*, Center for Computational Learning Systems Columbia University, New York, pp 573-580.
 39. Habash, N. & Rambow, O. 2006. 'MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects', in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, Sydney, pp 681-688.

40. Habash, N. & Sadat, F. 2006. 'Arabic Preprocessing Schemes for Statistical Machine Translation', in *Proceedings of the Human Language Technologies Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL) 2006*, New York.
41. Habash, N. & Roth, R. 2009. 'CATiB: The Columbia Arabic Treebank', in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Association for Computational Linguistics, Singapore, pp 221-224. DOI: 10.3115/1667583.1667651
42. Habash, N., Faraj, R. & Roth, R. 2009. 'Syntactic Annotation in the Columbia Arabic Treebank', in *Conference on Arabic Language Resources and Tools*, Cairo, pp 125-132.
43. HABASH, N.Y. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
44. Habash, N., Rambow, O. & Roth, R. 2010. 'MADA+TOKAN Manual (Current Version: MADA-3.0.1)', Center for Computational Learning Systems.
45. Hajič, J., Smrž, O., Zemánek, P., Šnaidauf, J. & Beška. 2004. 'Prague Arabic Dependency Treebank: Development in Data and Tools', in *Proceedings of NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, pp 110-117.
46. Hajič, J., Smrž, O., Buckwalter, T & Jin, H. 2005. 'Feature-based Tagger of Approximations of Functional Arabic Morphology', in Ma. Antonia Martí Montserrat Civit, Sandra Kübler (ed), *Proceedings of Treebanks and Linguistic Theories (TLT)*, Barcelona, pp 53-64.
47. Institute of Formal and Applied Linguistics. 2007-2010. Prague Arabic Dependency Treebank ++. Available from: <http://padt-online.blogspot.com/es/> [18 June 2016]
48. Jaafar, A., Mohamad Ali, N., Azman Mohd Noah, S., Smeaton, A.F., Bruza, P., Abu Bakar, Z., Jamil, N., Mohd Tengku Sembok, T. (eds.) 2014. *Information Retrieval Technology, 10th Asia Information Retrieval Societies Conference, AIRS 2014*, Springer International Publishing, Kuching.
49. JURAFSKY, D. & MARTIN, J.H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing*. Computational Linguistics, and Speech Recognition. International edn, Prentice Hall, Upper Saddle River, New Jersey.
50. Khoja, S. 2001. 'APT: Arabic Part-of-speech Tagger', Computing Department, Lancaster University.
51. Khoja, S., Garside, R. & Knowles, G. 2001. 'A tagset for the morphosyntactic tagging of Arabic', in *Proceedings of Corpus Linguistics 2001 Conference*, UCREL Technical Paper 13, Lancaster University, pp 341-353.
52. Khoufi, N., Aloulou, C. & Hadrich Belguith, L. 2015. 'Parsing Arabic using induced probabilistic context free grammar', in *International Journal of Speech Technology*, Vol. 19, online publication date: 4-Sept-2015, pp 313-323. DOI: 10.1007/s10772-015-9300-x
53. Kulick, S., Gabbard, R. & Marcus, M. 2006. 'Parsing the Arabic Treebank: Analysis and Improvements', in *Proceedings of the Treebanks and Linguistic Theories Conference*, Institute of Formal and Applied Linguistics, Prague, pp 31-42.

54. Kulick, S. 2010. 'Simultaneous Tokenization and Part-of-Speech Tagging for Arabic without a Morphological Analyzer', in *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, pp 342-347.
55. Maamouri, M., Bies, A., Buckwalter, T. & Jin, H. 2003. Arabic Treebank: Part 1 v 2.0. Available from: <https://catalog ldc.upenn.edu/LDC2003T06>. LDC Catalog number LDC2003T06
56. Maamouri, M., Bies, A., Buckwalter, T. & Mekki, W. 2004. 'The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus', in *NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, pp 102-109.
57. Maamouri, M., Bies, A., Buckwalter, T. & Jin, H. 2004. Arabic Treebank: Part 2 v 2.0. Available from: <https://catalog ldc.upenn.edu/LDC2004T02>. LDC Catalog number LDC2004T02
58. Maamouri, M., Bies, A., Buckwalter, T., Jin, H. & Mekki, W. 2004. Arabic Treebank: Part 3 (full corpus) v 2.0 (MPG + Syntactic Analysis). Available from: <https://catalog ldc.upenn.edu/LDC2005T20>. LDC Catalog number LDC2005T20
59. Maamouri, M., Bies, A., Buckwalter, T. & Jin, H. 2005. Arabic Treebank: Part 1 v 3.0. Available from: <http://www ldc.upenn.edu/>. LDC Catalog number LDC2005T02
60. Maamouri, M., Bies, A., Kulick, S., Gaddeche, F., Mekki, W., Krouna, S., Bouziri, B. & Zaghouni, W. 2010. Arabic Treebank: Part 1 v 4.1. Available from: <https://catalog ldc.upenn.edu/LDC2010T13>. LDC Catalog number LDC2010T13
61. Marques, N. & Lopes, J.G. 1996. 'Using Neural Nets for Portuguese Part-of-Speech Tagging', in *Proceedings of the Fifth International Conference on The Cognitive Science of Natural Language Processing*, Dublin City University.
62. Martí, M.A. & Llisterri, J. (eds) 2002. *Tratamiento del lenguaje natural*. UB 53 manuals, Edicions Universitat de Barcelona, Barcelona.
63. Martí, M.A. & Llisterri, J. (eds) 2004. *Tecnologías del texto y del habla*. Nº 72, Edicions Universitat de Barcelona, Barcelona.
64. Marton, Y., Habash, N. & Rambow, O. 2010. 'Improving Arabic Dependency Parsing with Lexical and Inflectional Morphological Features', in *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, Association for Computational Linguistics, Los Angeles, pp 13-21.
65. Menai, M.E.B. 2014. 'Word sense disambiguation using evolutionary algorithms: Application to Arabic language', in *Computers in Human Behavior*, No 41, College of Computer and Information Sciences, King Saud University, pp 92-103.
66. Métais, E., Meziane, F., Sararee, M., Sugumaran, V., Vadera, S. (eds.) 2013. *Natural Language Processing and Information Systems, 18th International Conference on Applications of Natural Language to Information Systems, NLDB 2013*, Springer, Salford.
67. Miller, G. 1995. 'Wordnet: A Lexical Database for English', in *Communications of the ACM*, Vol. 38, Nº 11, pp 39-41.

68. Mohamed, E. & Kübler, S. 2010. 'Is Arabic Part of Speech Tagging Feasible Without Word Segmentation?', in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, Los Angeles, pp 705-708.
69. Mokry, K. & Smrž, O. 2015. 'External tools not only for ArabTeX Documents', Faculty of Mathematics and Physics, Charles University, Prague.
70. Olde, B.A., Hoener, J., Chipman, P., Graesser, A.C. & Tutoring Research Group, 1999. 'A Connectionist Model for Part of Speech Tagging', in *Proceedings of the 12th International Florida Artificial Intelligence Research Society Conference*, Menlo Park, pp 172-176.
71. Roth, R., Rambow, O., Habash, N., Diab, M. & Rudin, C. 2008. 'Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking', in *Proceedings of ACL-08: HLT, Short Papers (Companion Volume)*, Ohio, pp 117-120.
72. Smrž, O & Zemánek, P. 2002. 'Sherds from an Arabic Treebanking Mosaic', in *Prague Bulletin of Mathematical Linguistics*, N° 78, pp 63-76.
73. Smrž, O & Pajas, P. 2004. 'MorphoTrees of Arabic and Their Annotation in the TrEd Environment', in *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, pp 38-41.
74. Smrž, O. (2016), Bielický, V. (2014), Buckwalter, T. (2002). ElixirFM Online Interface. Available from: <http://quest.ms.mff.cuni.cz/cgi-bin/elixir/index.fcgi> [18 June 2016]
75. Souidi, A., Bosh, A.V.D & Neumann, G. (eds.) 2007. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Text, Speech and Language technology, Vol. 38, Springer, Netherlands.
76. Tait, J.I. (ed) 2005. *Charting a New Course: Natural Language Processing and Information Retrieval. Essays in Honour of Karen Spärck Jones*. Springer, Netherlands.
77. Tounsi, L., Attia, M. & van Genabith, J. 2009. 'Parsing Arabic using Treebank-based LFG resources', in *Proceedings of the LFG09 Conference*, CSLI Publications, pp 583-586.
78. Turki Khemakhem, I., Jamoussi, S. & Ben Hamadou, A. 2010. 'Arabic morpho-syntactic feature disambiguation in a translation context', in *Proceedings of SSST-4, Fourth Workshop on Syntax and Structure in Statistical Translation*, COLING 2010, Beijing, pp 61-65.
79. Žabokrtský, Z. & Smrž, O. 2003. 'Arabic syntactic trees: from constituency to dependency', in *EACL '03 Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, Vol. 2, Association for Computational Linguistics, Stroudsburg, pp 183-186

Annex 1: List of abbreviations

Abbreviation	Meaning
ACC	Accusative
ADJ	Adjective
ADJP	Adjective Phrase
ADVP	Adverb Phrase
ALMORGEANA / ALMOR	Arabic Lexeme-Based Morphological Generation and Analysis
BAMA	Buckwalter Arabic Morphology Analyzer
BPC	Base Phrase Chunking
CATiB	The Columbia Arabic Treebank
CONJP	Conjunction Phrase
DARPA	Defense Advanced Research Projects Agency
DEF	Definite
DET / DT	Determiner
DS	Dependency Structure
ERTS	Extended Reduced Tag Set
ETS	Educated Text Stemmer
FEM	Feminine
FSM	Finite State Machine
HMM	Hidden Markov Model
IR	Information Retrieval
IE	Information Extraction
INTJ	Interjection
I O B	Inside the chunk Outside the chunk Beginning of the chunk
JJ	Adjective (regardles the inflection)
LDC	Linguistic Data Consortium

MADA	Morphological Analysis and Disambiguation for Arabic
MAGEAD	Morphological Analyzer and Generator for the Arabic Dialects
MBC	Morphological Behaviour Class
ML	Machine Learning
MSA	Modern Standard Arabic
MT	Machine Translation
NLP	Natural Language Processing
NLUSA	Natural Language Understanding System for Arabic
NP	Noun phrase
NSUFF	Nominal suffix
PATB	The Penn Arabic Treebank
PADT	The Prague Arabic Dependency Treebank
PoS	Part-of-Speech
POSS	Possessive
PP	Prepositional Phrase
PREP	Preposition
PRON	Pronoun
PRT	Particle
PS	Phrase Structure
QA	Question Answering
RTS	Reduced Tag Set
SAMA	Standard Arabic Morphological Analyzer
SBAR	Clause introduced by a subordinating conjunction.
SG	Singular
SVP	Support Vector Machine
VP	Verb Phrase
1s	First Person Singular

Annex 2: Arabic Unicode: U+0600 to U+06FF

Unicode, 1995-2015. Arabic Unicode, Range 0600-06FF, Unicode Inc.

Arabic

Range: 0600–06FF

This file contains an excerpt from the character code tables and list of character names for *The Unicode Standard, Version 8.0*

This file may be changed at any time without notice to reflect errata or other updates to the Unicode Standard. See <http://www.unicode.org/errata/> for an up-to-date list of errata.

See <http://www.unicode.org/charts/> for access to a complete list of the latest character code charts.
See <http://www.unicode.org/charts/PDF/Unicode-8.0/> for charts showing only the characters added in Unicode 8.0.
See <http://www.unicode.org/Public/8.0.0/charts/> for a complete archived file of character code charts for Unicode 8.0.

Disclaimer

These charts are provided as the online reference to the character contents of the Unicode Standard, Version 8.0 but do not provide all the information needed to fully support individual scripts using the Unicode Standard. For a complete understanding of the use of the characters contained in this file, please consult the appropriate sections of The Unicode Standard, Version 8.0, online at <http://www.unicode.org/versions/Unicode8.0.0/>, as well as Unicode Standard Annexes #9, #11, #14, #15, #24, #29, #31, #34, #38, #41, #42, #44, and #45, the other Unicode Technical Reports and Standards, and the Unicode Character Database, which are available online.

See <http://www.unicode.org/ucd/> and <http://www.unicode.org/reports/>

A thorough understanding of the information contained in these additional sources is required for a successful implementation.

Fonts

The shapes of the reference glyphs used in these code charts are not prescriptive. Considerable variation is to be expected in actual fonts. The particular fonts used in these charts were provided to the Unicode Consortium by a number of different font designers, who own the rights to the fonts.

See <http://www.unicode.org/charts/fonts.html> for a list.

Terms of Use

You may freely use these code charts for personal or internal business uses only. You may not incorporate them either wholly or in part into any product or publication, or otherwise distribute them without express written permission from the Unicode Consortium. However, you may provide links to these charts.

The fonts and font data used in production of these code charts may NOT be extracted, or used in any other way in any product or publication, without permission or license granted by the typeface owner(s).






The Unicode Consortium is not liable for errors or omissions in this file or the standard itself. Information on characters added to the Unicode Standard since the publication of the most recent version of the Unicode Standard, as well as on characters currently being considered for addition to the Unicode Standard can be found on the Unicode web site.

See <http://www.unicode.org/pending/pending.html> and <http://www.unicode.org/alloc/Pipeline.html>.


Copyright © 1991-2015 Unicode, Inc. All rights reserved.

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0		ي	ذ	-	◌ِ	◌ْ	◌ُ	ي	ذ	غ	گ	ه	ي	◌ِ	◌ْ	◌ُ
	0600	0610	0620	0630	0640	0650	0660	0670	0680	0690	06A0	06B0	06C0	06D0	06E0	06F0
1		ع	ر	ف	◌ِ	◌ْ	◌ُ	أ	ح	ر	ف	گ	ه	ي	◌ِ	◌ْ
	0601	0611	0621	0631	0641	0651	0661	0671	0681	0691	06A1	06B1	06C1	06D1	06E1	06F1
2		ح	آ	ز	ق	◌ِ	◌ْ	أ	خ	ز	ب	گ	ه	ي	◌ِ	◌ْ
	0602	0612	0622	0632	0642	0652	0662	0672	0682	0692	06A2	06B2	06C2	06D2	06E2	06F2
3		ض	أ	س	ك	◌ِ	◌ْ	أ	ح	ر	ب	گ	ه	ي	◌ِ	◌ْ
	0603	0613	0623	0633	0643	0653	0663	0673	0683	0693	06A3	06B3	06C3	06D3	06E3	06F3
4		ي	ؤ	ش	ل	◌ِ	◌ْ	ع	ج	ر	ف	گ	ه	-	◌ِ	◌ْ
	0604	0614	0624	0634	0644	0654	0664	0674	0684	0694	06A4	06B4	06C4	06D4	06E4	06F4
5		ط	إ	ص	م	◌ِ	◌ْ	أ	خ	ر	ب	ل	و	ه	ر	ه
	0605	0615	0625	0635	0645	0655	0665	0675	0685	0695	06A5	06B5	06C5	06D5	06E5	06F5
6		ل	ي	ض	ن	◌ِ	◌ْ	ؤ	ح	ر	ف	ل	ؤ	◌ِ	◌ْ	◌ُ
	0606	0616	0626	0636	0646	0656	0666	0676	0686	0696	06A6	06B6	06C6	06D6	06E6	06F6
7		ر	ا	ط	ه	◌ِ	◌ْ	ؤ	ح	ر	ف	ل	ؤ	◌ِ	◌ْ	◌ُ
	0607	0617	0627	0637	0647	0657	0667	0677	0687	0697	06A7	06B7	06C7	06D7	06E7	06F7
8		ي	ب	ظ	و	◌ِ	◌ْ	أ	ذ	ر	ف	ل	ؤ	◌ِ	◌ْ	◌ُ
	0608	0618	0628	0638	0648	0658	0668	0678	0688	0698	06A8	06B8	06C8	06D8	06E8	06F8
9		ي	ة	ع	ي	◌ِ	◌ْ	ط	ر	ر	ك	ب	ؤ	◌ِ		ه
	0609	0619	0629	0639	0649	0659	0669	0679	0689	0699	06A9	06B9	06C9	06D9	06E9	06F9
A		ي	ت	غ	ي	◌ِ	◌ْ	ن	ب	ب	ك	ر	ق	◌ِ	◌ْ	بش
	060A	061A	062A	063A	064A	065A	066A	067A	068A	069A	06AA	06BA	06CA	06DA	06EA	06FA
B		ف	؛	ث	ك	◌ِ	◌ْ	ر	ب	ب	ك	ر	ق	◌ِ	◌ْ	ض
	060B	061B	062B	063B	064B	065B	066B	067B	068B	069B	06AB	06BB	06CB	06DB	06EB	06FB
C	،		ج	ك	◌ِ	◌ْ	،	ت	ذ	ث	ك	ر	ق	◌ِ	◌ْ	غ
	060C	061C	062C	063C	064C	065C	066C	067C	068C	069C	06AC	06BC	06CC	06DC	06EC	06FC
D	،		ح	ي	◌ِ	◌ْ	*	ت	د	ص	ك	ن	م		◌ِ	ه
	060D		062D	063D	064D	065D	066D	067D	068D	069D	06AD	06BD	06CD	06DD	06ED	06FD
E	م	ه	خ	ي	◌ِ	◌ْ	ب	پ	ذ	ض	ك	ه	ي		ذ	م
	060E	061E	062E	063E	064E	065E	066E	067E	068E	069E	06AE	06BE	06CE	06DE	06EE	06FE
F	ع	؟	د	ث	◌ِ	◌ْ	ف	ت	ذ	ظ	گ	ف	ؤ	◌ِ	ر	ه
	060F	061F	062F	063F	064F	065F	066F	067F	068F	069F	06AF	06BF	06CF	06DF	06EF	06FF





Subtending marks

- 0600  ARABIC NUMBER SIGN
 0601  ARABIC SIGN SANAH
 0602  ARABIC FOOTNOTE MARKER
 0603  ARABIC SIGN SAFHA
 0604  ARABIC SIGN SAMVAT
 • used for writing Samvat era dates in Urdu


Supertending mark

- 0605  ARABIC NUMBER MARK ABOVE
 • may be used with Coptic Epact numbers



Radix symbols

- 0606  ARABIC-INDIC CUBE ROOT
 → 221B  cube root
 0607  ARABIC-INDIC FOURTH ROOT
 → 221C  fourth root


Letterlike symbol

- 0608  ARABIC RAY





Punctuation

- 0609  ARABIC-INDIC PER MILLE SIGN
 → 2030 ‰ per mille sign
 060A  ARABIC-INDIC PER TEN THOUSAND SIGN
 → 2031 ‰ per ten thousand sign

Currency sign

- 060B  AFGHANI SIGN

Punctuation

- 060C  ARABIC COMMA
 • also used with Thaana and Syriac in modern text
 → 002C , comma
 → 2E32  turned comma
 → 2E41  reversed comma
 060D  ARABIC DATE SEPARATOR


Poetic marks

- 060E  ARABIC POETIC VERSE SIGN
 060F  ARABIC SIGN MISRA


Honorifics

- 0610  ARABIC SIGN SALLALLAHOU ALAYHE WASSALLAM
 • represents sallallahu alayhe wasallam “may God’s peace and blessings be upon him”
 0611  ARABIC SIGN ALAYHE ASSALLAM
 • represents alayhe assalam “upon him be peace”
 0612  ARABIC SIGN RAHMATULLAH ALAYHE
 • represents rahmatullah alayhe “may God have mercy upon him”
 0613  ARABIC SIGN RADI ALLAHOU ANHU
 • represents radi allahu ‘anhu “may God be pleased with him”
 0614  ARABIC SIGN TAKHALLUS
 • sign placed over the name or nom-de-plume of a poet, or in some writings used to mark all proper names








Koranic annotation sign

- 0615  ARABIC SMALL HIGH TAH
 • marks a recommended pause position in some Korans published in Iran and Pakistan
 • should not be confused with the small TAH sign used as a diacritic for some letters such as 0679 ت




Extended Arabic mark

- 0616  ARABIC SMALL HIGH LIGATURE ALEF WITH LAM WITH YE
 • early Persian


Koranic annotation signs

- 0617  ARABIC SMALL HIGH ZAIN
 0618  ARABIC SMALL FATHA
 • should not be confused with 064E  FATHA
 0619  ARABIC SMALL DAMMA
 • should not be confused with 064F  DAMMA
 061A  ARABIC SMALL KASRA
 • should not be confused with 0650  KASRA




Punctuation

- 061B  ARABIC SEMICOLON
 • also used with Thaana and Syriac in modern text
 → 003B ; semicolon
 → 204F  reversed semicolon
 → 2E35  turned semicolon

Format character

- 061C  ARABIC LETTER MARK
 • commonly abbreviated ALM
 → 200F  right-to-left mark

Punctuation





























- 061E  ARABIC TRIPLE DOT PUNCTUATION MARK
 061F  ARABIC QUESTION MARK
 • also used with Thaana and Syriac in modern text
 → 003F ? question mark
 → 2E2E  reversed question mark

Addition for Kashmiri

- 0620  ARABIC LETTER KASHMIRI YEH

Based on ISO 8859-6

Arabic letter names follow romanization conventions derived from ISO 8859-6. These differ from the Literary Arabic pronunciation of the letter names. For example, 0628 ARABIC LETTER BEH has a Literary Arabic pronunciation of ba’.

- 0621  ARABIC LETTER HAMZA
 → 02BE  modifier letter right half ring
 0622  ARABIC LETTER ALEF WITH MADDA ABOVE
 ≡ 0627  0653 
 0623  ARABIC LETTER ALEF WITH HAMZA ABOVE
 ≡ 0627  0654 
 0624  ARABIC LETTER WAW WITH HAMZA ABOVE
 ≡ 0648  0654 
 0625  ARABIC LETTER ALEF WITH HAMZA BELOW
 ≡ 0627  0655 
 0626  ARABIC LETTER YEH WITH HAMZA ABOVE
 ≡ 064A  0654 
 0627  ARABIC LETTER ALEF
 0628  ARABIC LETTER BEH
 0629  ARABIC LETTER TEH MARBUTA
 062A  ARABIC LETTER TEH
 062B  ARABIC LETTER THEH
 062C  ARABIC LETTER JEEM
 062D  ARABIC LETTER HAH
 062E  ARABIC LETTER KHAH
 062F  ARABIC LETTER DAL
 0630  ARABIC LETTER THAL
 0631  ARABIC LETTER REH

0632	ز	ARABIC LETTER ZAIN
0633	س	ARABIC LETTER SEEN
0634	ش	ARABIC LETTER SHEEN
0635	ص	ARABIC LETTER SAD
0636	ض	ARABIC LETTER DAD
0637	ط	ARABIC LETTER TAH
0638	ظ	ARABIC LETTER ZAH
0639	ع	ARABIC LETTER AIN
		→ 01B9 ع latin small letter ezh reversed
		→ 02BF ع modifier letter left half ring
063A	غ	ARABIC LETTER GHAIN

Additions for early Persian and Azerbaijani

063B	ک	ARABIC LETTER KEHEH WITH TWO DOTS ABOVE
063C	ک	ARABIC LETTER KEHEH WITH THREE DOTS BELOW
063D	ئ	ARABIC LETTER FARSI YEH WITH INVERTED V
		• Azerbaijani
063E	ئ	ARABIC LETTER FARSI YEH WITH TWO DOTS ABOVE
063F	ئ	ARABIC LETTER FARSI YEH WITH THREE DOTS ABOVE

Based on ISO 8859-6

0640	-	ARABIC TATWEEL
		= kashida
		• inserted to stretch characters or to carry tashkil with no base letter
		• also used with Mandaic, Manichaean, Psalter Pahlavi, and Syriac
0641	ف	ARABIC LETTER FEH
0642	ق	ARABIC LETTER QAF
0643	ك	ARABIC LETTER KAF
0644	ل	ARABIC LETTER LAM
0645	م	ARABIC LETTER MEEM
0646	ن	ARABIC LETTER NOON
0647	ه	ARABIC LETTER HEH
0648	و	ARABIC LETTER WAW
0649	ى	ARABIC LETTER ALEF MAKSURA
		• represents YEH-shaped dual-joining letter with no dots in any positional form
		• not intended for use in combination with 0654 ّ
		→ 0626 ئ arabic letter yeh with hamza above
064A	ي	ARABIC LETTER YEH
		• loses its dots when used in combination with 0654 ّ
		• retains its dots when used in combination with other combining marks
		→ 08A8 ئي arabic letter yeh with two dots below and hamza above

Tashkil from ISO 8859-6

064B	◌َ	ARABIC FATHATAN
064C	◌ِ	ARABIC DAMMATAN
		• a common alternative form is written as two intertwined dammas, one of which is turned 180 degrees
064D	◌ُ	ARABIC KASRATAN
064E	◌ْ	ARABIC FATHA
064F	◌ِ	ARABIC DAMMA
0650	◌ُ	ARABIC KASRA
0651	◌ْ	ARABIC SHADDA

0652	◌◌◌	ARABIC SUKUN
		• marks absence of a vowel after the base consonant
		• used in some Korans to mark a long vowel as ignored
		• can have a variety of shapes, including a circular one and a shape that looks like ّ◌◌◌
		→ 06E1 ◌◌◌ arabic small high dotless head of khah

Combining maddah and hamza

0653	◌ْ◌◌◌	ARABIC MADDAH ABOVE
0654	◌◌◌◌◌	ARABIC HAMZA ABOVE
		• restricted to hamza and ezafe semantics
		• is not used as a diacritic to form new letters
0655	◌◌◌◌◌◌	ARABIC HAMZA BELOW

Other combining marks

0656	◌◌◌◌◌◌	ARABIC SUBSCRIPT ALEF
0657	◌◌◌◌◌◌◌	ARABIC INVERTED DAMMA
		= ulta pesh
		• Kashmiri, Urdu
0658	◌◌◌◌◌◌◌◌	ARABIC MARK NOON GHUNNA
		• Baluchi
		• indicates nasalization in Urdu
0659	◌◌◌◌◌◌◌◌◌	ARABIC ZWARAKAY
		• Pashto
065A	◌◌◌◌◌◌◌◌◌◌	ARABIC VOWEL SIGN SMALL V ABOVE
		• African languages
065B	◌◌◌◌◌◌◌◌◌◌◌	ARABIC VOWEL SIGN INVERTED SMALL V ABOVE
		• African languages
065C	◌◌◌◌◌◌◌◌◌◌◌◌	ARABIC VOWEL SIGN DOT BELOW
		• African languages
065D	◌◌◌◌◌◌◌◌◌◌◌◌◌	ARABIC REVERSED DAMMA
		• African languages
065E	◌◌◌◌◌◌◌◌◌◌◌◌◌◌	ARABIC FATHA WITH TWO DOTS
		• Kalami
065F	◌◌◌◌◌◌◌◌◌◌◌◌◌◌◌	ARABIC WAVY HAMZA BELOW
		• Kashmiri

Arabic-Indic digits

These digits are used with Arabic proper; for languages of Iran, Afghanistan, Pakistan, and India, see the Eastern Arabic-Indic digits at 06F0-06F9.

0660	٠	ARABIC-INDIC DIGIT ZERO
0661	١	ARABIC-INDIC DIGIT ONE
0662	٢	ARABIC-INDIC DIGIT TWO
0663	٣	ARABIC-INDIC DIGIT THREE
0664	٤	ARABIC-INDIC DIGIT FOUR
0665	٥	ARABIC-INDIC DIGIT FIVE
0666	٦	ARABIC-INDIC DIGIT SIX
0667	٧	ARABIC-INDIC DIGIT SEVEN
0668	٨	ARABIC-INDIC DIGIT EIGHT
0669	٩	ARABIC-INDIC DIGIT NINE

Punctuation

066A	٪	ARABIC PERCENT SIGN
		→ 0025 % percent sign
066B	٫	ARABIC DECIMAL SEPARATOR
066C	’	ARABIC THOUSANDS SEPARATOR
		→ 0027 ’ apostrophe
		→ 2019 ’ right single quotation mark
066D	*	ARABIC FIVE POINTED STAR
		• appearance rather variable
		→ 002A * asterisk

Archaic letters

066E ٲ ARABIC LETTER DOTLESS BEH

066F ٴ ARABIC LETTER DOTLESS QAF

Point0670 ٲ ARABIC LETTER SUPERScript ALEF
• actually a vowel sign, despite the name**Extended Arabic letters**

0671 ٲ ARABIC LETTER ALEF WASLA

• Koranic Arabic

0672 ٲ ARABIC LETTER ALEF WITH WAVY HAMZA ABOVE

• Baluchi, Kashmiri

Deprecated letter

0673 ٲ ARABIC LETTER ALEF WITH WAVY HAMZA BELOW

• Kashmiri

• this character is deprecated and its use is strongly discouraged

• use the sequence 0627 ٲ 065F ٲ instead

Extended Arabic letters

0674 ٲ ARABIC LETTER HIGH HAMZA

• Kazakh

• forms digraphs

0675 ٲ ARABIC LETTER HIGH HAMZA ALEF

• Kazakh

≈ 0627 ٲ 0674 ٲ

0676 ٲ ARABIC LETTER HIGH HAMZA WAW

• Kazakh

≈ 0648 ٲ 0674 ٲ

0677 ٲ ARABIC LETTER U WITH HAMZA ABOVE

• Kazakh

≈ 06C7 ٲ 0674 ٲ

0678 ٲ ARABIC LETTER HIGH HAMZA YEH

• Kazakh

≈ 064A ٲ 0674 ٲ

0679 ٲ ARABIC LETTER TTEH

• Urdu

067A ٲ ARABIC LETTER TTEHEH

• Sindhi

067B ٲ ARABIC LETTER BEEH

• Sindhi

067C ٲ ARABIC LETTER TEH WITH RING

• Pashto

067D ٲ ARABIC LETTER TEH WITH THREE DOTS ABOVE DOWNWARDS

• Sindhi

067E ٲ ARABIC LETTER PEH

• Persian, Urdu, ...

067F ٲ ARABIC LETTER TEHEH

• Sindhi

0680 ٲ ARABIC LETTER BEHEH

• Sindhi

0681 ٲ ARABIC LETTER HAH WITH HAMZA ABOVE

• Pashto

• represents the phoneme /dz/

0682 ٲ ARABIC LETTER HAH WITH TWO DOTS VERTICAL ABOVE

• not used in modern Pashto

0683 ٲ ARABIC LETTER NYEH

• Sindhi

0684 ٲ ARABIC LETTER DYEH

• Sindhi

0685 ٲ ARABIC LETTER HAH WITH THREE DOTS ABOVE

• Pashto, Khwarazmian

• represents the phoneme /ts/ in Pashto

0686 ٲ ARABIC LETTER TCHEH

• Persian, Urdu, ...

0687 ٲ ARABIC LETTER TCHEHEH

• Sindhi

0688 ٲ ARABIC LETTER DDAL

• Urdu

0689 ٲ ARABIC LETTER DAL WITH RING

• Pashto

068A ٲ ARABIC LETTER DAL WITH DOT BELOW

• Sindhi, early Persian

068B ٲ ARABIC LETTER DAL WITH DOT BELOW AND SMALL TAH

• Lahnda

068C ٲ ARABIC LETTER DAHAL

• Sindhi

068D ٲ ARABIC LETTER DDAHAL

• Sindhi

068E ٲ ARABIC LETTER DUL

• older shape for DUL, now obsolete in Sindhi

• Burushaski

068F ٲ ARABIC LETTER DAL WITH THREE DOTS ABOVE DOWNWARDS

• Sindhi

• current shape used for DUL

0690 ٲ ARABIC LETTER DAL WITH FOUR DOTS ABOVE

• old Urdu, not in current use

0691 ٲ ARABIC LETTER RREH

• Urdu

0692 ٲ ARABIC LETTER REH WITH SMALL V

• Kurdish

0693 ٲ ARABIC LETTER REH WITH RING

• Pashto

0694 ٲ ARABIC LETTER REH WITH DOT BELOW

• Kurdish, early Persian

0695 ٲ ARABIC LETTER REH WITH SMALL V BELOW

• Kurdish

0696 ٲ ARABIC LETTER REH WITH DOT BELOW AND DOT ABOVE

• Pashto

0697 ٲ ARABIC LETTER REH WITH TWO DOTS ABOVE

• Dargwa

0698 ٲ ARABIC LETTER JEH

• Persian, Urdu, ...

0699 ٲ ARABIC LETTER REH WITH FOUR DOTS ABOVE

• Sindhi

069A ٲ ARABIC LETTER SEEN WITH DOT BELOW AND DOT ABOVE

• Pashto

069B ٲ ARABIC LETTER SEEN WITH THREE DOTS BELOW

• early Persian

069C ٲ ARABIC LETTER SEEN WITH THREE DOTS BELOW AND THREE DOTS ABOVE

• Moroccan Arabic

069D ٲ ARABIC LETTER SAD WITH TWO DOTS BELOW

• Turkic

069E ٲ ARABIC LETTER SAD WITH THREE DOTS ABOVE

• Berber, Burushaski

069F	ظ	ARABIC LETTER TAH WITH THREE DOTS ABOVE	06BD	ن	ARABIC LETTER NOON WITH THREE DOTS ABOVE
		• old Hausa			• old Malay
06A0	غ	ARABIC LETTER AIN WITH THREE DOTS ABOVE	06BE	ه	ARABIC LETTER HEH DOACHASHMEE
		• old Malay			• forms aspirate digraphs in Urdu and other languages of South Asia
06A1	ف	ARABIC LETTER DOTLESS FEH			• represents the glottal fricative /h/ in Uighur
		• Adighe	06BF	ح	ARABIC LETTER TCHEH WITH DOT ABOVE
06A2	ب	ARABIC LETTER FEH WITH DOT MOVED BELOW	06C0	ة	ARABIC LETTER HEH WITH YEH ABOVE
		• Maghrib Arabic			= arabic letter hamzah on ha (1.0)
06A3	ب	ARABIC LETTER FEH WITH DOT BELOW			• for ezafe, use 0654 ة over the language-appropriate base letter
		• Ingush			• actually a ligature, not an independent letter
06A4	ث	ARABIC LETTER VEH			≡ 06D5 • 0654 ة
		• Middle Eastern Arabic for foreign words	06C1	ـ	ARABIC LETTER HEH GOAL
		• Kurdish, Khwarazmian, early Persian			• Urdu
06A5	پ	ARABIC LETTER FEH WITH THREE DOTS BELOW	06C2	ـ	ARABIC LETTER HEH GOAL WITH HAMZA ABOVE
		• North African Arabic for foreign words			• Urdu
06A6	ق	ARABIC LETTER PEHEH			• actually a ligature, not an independent letter
		• Sindhi			≡ 06C1 ـ 0654 ة
06A7	ق	ARABIC LETTER QAF WITH DOT ABOVE	06C3	ـ	ARABIC LETTER TEH MARBUTA GOAL
		• Maghrib Arabic, Uighur			• Urdu
06A8	ق	ARABIC LETTER QAF WITH THREE DOTS ABOVE	06C4	و	ARABIC LETTER WAW WITH RING
		• Tunisian Arabic			• Kashmiri
06A9	ک	ARABIC LETTER KEHEH	06C5	و	ARABIC LETTER KIRGHIZ OE
		• Persian, Urdu, ...			• Kirghiz
06AA	ڪ	ARABIC LETTER SWASH KAF	06C6	ؤ	ARABIC LETTER OE
		• represents a letter distinct from Arabic KAF (0643 ك) in Sindhi			• Uighur, Kurdish, Kazakh, Azerbaijani
06AB	ک	ARABIC LETTER KAF WITH RING	06C7	ۇ	ARABIC LETTER U
		• Pashto			• Kirghiz, Azerbaijani
		• may appear like an Arabic KAF (0643 ك) with a ring below the base	06C8	ۇ	ARABIC LETTER YU
06AC	ك	ARABIC LETTER KAF WITH DOT ABOVE			• Uighur
		• old Malay	06C9	ؤ	ARABIC LETTER KIRGHIZ YU
06AD	ڱ	ARABIC LETTER NG			• Kazakh, Kirghiz
		• Uighur, Kazakh, old Malay, early Persian, ...	06CA	ق	ARABIC LETTER WAW WITH TWO DOTS ABOVE
06AE	ك	ARABIC LETTER KAF WITH THREE DOTS BELOW			• Kurdish
		• Berber, early Persian	06CB	ؤ	ARABIC LETTER VE
06AF	گ	ARABIC LETTER GAF			• Uighur, Kazakh
		• Persian, Urdu, ...	06CC	ی	ARABIC LETTER FARSI YEH
06B0	گ	ARABIC LETTER GAF WITH RING			• Arabic, Persian, Urdu, Kashmiri, ...
		• Lahnda			• initial and medial forms of this letter have dots
06B1	گھ	ARABIC LETTER NGOEH			→ 0649 ی arabic letter alef maksura
		• Sindhi			→ 064A ی arabic letter yeh
06B2	گھ	ARABIC LETTER GAF WITH TWO DOTS BELOW	06CD	ی	ARABIC LETTER YEH WITH TAIL
		• not used in Sindhi			• Pashto, Sindhi
06B3	گھ	ARABIC LETTER GUEH	06CE	ی	ARABIC LETTER YEH WITH SMALL V
		• Sindhi			• Kurdish
06B4	گھ	ARABIC LETTER GAF WITH THREE DOTS ABOVE	06CF	و	ARABIC LETTER WAW WITH DOT ABOVE
		• not used in Sindhi	06D0	ی	ARABIC LETTER E
06B5	ل	ARABIC LETTER LAM WITH SMALL V			• Pashto, Uighur
		• Kurdish			• used as the letter bbeh in Sindhi
06B6	ل	ARABIC LETTER LAM WITH DOT ABOVE	06D1	ی	ARABIC LETTER YEH WITH THREE DOTS BELOW
		• Kurdish			• old Malay
06B7	ل	ARABIC LETTER LAM WITH THREE DOTS ABOVE	06D2	ے	ARABIC LETTER YEH BARREE
		• Kurdish			• Urdu
06B8	لپ	ARABIC LETTER LAM WITH THREE DOTS BELOW	06D3	ے	ARABIC LETTER YEH BARREE WITH HAMZA ABOVE
06B9	ن	ARABIC LETTER NOON WITH DOT BELOW			• Urdu
06BA	ن	ARABIC LETTER NOON GHUNNA			• actually a ligature, not an independent letter
		• Urdu, archaic Arabic			≡ 06D2 ے 0654 ة
		• dotless in all four contextual forms			
06BB	ن	ARABIC LETTER RNOON			
		• Sindhi			
06BC	ن	ARABIC LETTER NOON WITH RING			
		• Pashto			

Punctuation

06D4 - ARABIC FULL STOP
• Urdu

Extended Arabic letter

06D5 • ARABIC LETTER AE
• Uighur, Kazakh, Kirghiz

Koranic annotation signs

06D6 ◌ ARABIC SMALL HIGH LIGATURE SAD WITH LAM WITH ALEF MAKSURA

06D7 ◌ ARABIC SMALL HIGH LIGATURE QAF WITH LAM WITH ALEF MAKSURA

06D8 ◌ ARABIC SMALL HIGH MEEM INITIAL FORM

06D9 ◌ ARABIC SMALL HIGH LAM ALEF

06DA ◌ ARABIC SMALL HIGH JEEM

06DB ◌ ARABIC SMALL HIGH THREE DOTS

06DC ◌ ARABIC SMALL HIGH SEEN

06DD ◌ ARABIC END OF AYAH

06DE ◌ ARABIC START OF RUB EL HIZB

06DF ◌ ARABIC SMALL HIGH ROUNDED ZERO
• smaller than the typical circular shape used for 0652 ◌

06E0 ◌ ARABIC SMALL HIGH UPRIGHT RECTANGULAR ZERO

06E1 ◌ ARABIC SMALL HIGH DOTLESS HEAD OF KHAH = Arabic jazm
• presentation form of 0652 ◌, using font technology to select the variant is preferred
• used in some Korans to mark absence of a vowel
→ 0652 ◌ arabic sukun

06E2 ◌ ARABIC SMALL HIGH MEEM ISOLATED FORM

06E3 ◌ ARABIC SMALL LOW SEEN

06E4 ◌ ARABIC SMALL HIGH MADDA
• typically used with 06E5 ◌, 06E6 ◌, 06E7 ◌, and 08F3 ◌

06E5 ◌ ARABIC SMALL WAW
→ 08F3 ◌ arabic small high waw

06E6 ◌ ARABIC SMALL YEH

06E7 ◌ ARABIC SMALL HIGH YEH

06E8 ◌ ARABIC SMALL HIGH NOON

06E9 ◌ ARABIC PLACE OF SAJDAH
• there is a range of acceptable glyphs for this character

06EA ◌ ARABIC EMPTY CENTRE LOW STOP

06EB ◌ ARABIC EMPTY CENTRE HIGH STOP

06EC ◌ ARABIC ROUNDED HIGH STOP WITH FILLED CENTRE

06ED ◌ ARABIC SMALL LOW MEEM

Extended Arabic letters for Parkari

06EE ◌ ARABIC LETTER DAL WITH INVERTED V

06EF ◌ ARABIC LETTER REH WITH INVERTED V
• also used in early Persian

Eastern Arabic-Indic digits

These digits are used with Arabic-script languages of Iran, Pakistan, and India (Persian, Sindhi, Urdu, etc.). For details of variations in preferred glyphs, see the block description for the Arabic script.

06F0 ◌ EXTENDED ARABIC-INDIC DIGIT ZERO

06F1 ◌ EXTENDED ARABIC-INDIC DIGIT ONE

06F2 ◌ EXTENDED ARABIC-INDIC DIGIT TWO

06F3 ◌ EXTENDED ARABIC-INDIC DIGIT THREE

06F4 ◌ EXTENDED ARABIC-INDIC DIGIT FOUR
• Persian has a different glyph than Sindhi and Urdu

06F5 ◌ EXTENDED ARABIC-INDIC DIGIT FIVE
• Persian, Sindhi, and Urdu share glyph different from Arabic

06F6 ◌ EXTENDED ARABIC-INDIC DIGIT SIX
• Persian, Sindhi, and Urdu have glyphs different from Arabic

06F7 ◌ EXTENDED ARABIC-INDIC DIGIT SEVEN
• Urdu and Sindhi have glyphs different from Arabic

06F8 ◌ EXTENDED ARABIC-INDIC DIGIT EIGHT

06F9 ◌ EXTENDED ARABIC-INDIC DIGIT NINE

Extended Arabic letters

06FA ◌ ARABIC LETTER SHEEN WITH DOT BELOW

06FB ◌ ARABIC LETTER DAD WITH DOT BELOW

06FC ◌ ARABIC LETTER GHAIN WITH DOT BELOW

Signs for Sindhi

06FD ◌ ARABIC SIGN SINDHI AMPERSAND

06FE ◌ ARABIC SIGN SINDHI POSTPOSITION MEN

Extended Arabic letter for Parkari

06FF ◌ ARABIC LETTER HEH WITH INVERTED V

Annex 3: The Khoja tag set

Khoja, S., Garside, R. & Knowles, G. 2001. 'A tagset for the morphosyntactic tagging of Arabic', in Proceedings of Corpus Linguistics 2001 Conference , UCREL Technical Paper 13, Lancaster University, pp 341353.

Tag	Description of word category	Example (Arabic)	Transcription	Translation
NCSgMNI	Singular, masculine, nominative, indefinite common noun	كتاب	<i>kitabun</i>	book
NCSgMAI	Singular, masculine, accusative, indefinite common noun	كتابا	<i>kitabān</i>	book
NCSgMGI	Singular, masculine, genitive, indefinite common noun	كتابي	<i>kitabīn</i>	book
NCSgMND	Singular, masculine, nominative, definite common noun	الكتاب	<i>alkitabu</i>	the book
NCSgMAD	Singular, masculine, accusative, definite common noun	الكتابا	<i>alkitaba</i>	the book
NCSgMGD	Singular, masculine, genitive, definite common noun	الكتابي	<i>alkitabī</i>	the book
NCSgFNI	Singular, feminine, nominative, indefinite common noun	مدرسة	<i>madrasatun</i>	school
NCSgFAI	Singular, feminine, accusative, indefinite common noun	مدرسةا	<i>madrasatan</i>	school
NCSgFGI	Singular, feminine, genitive, indefinite common noun	مدرسةي	<i>madrasatīn</i>	school
NCSgFND	Singular, feminine, nominative, definite common noun	المدرسة	<i>al-madrasatu</i>	the school

NCSgFAD	Singular, feminine, accusative, definite common noun	المدرسة	<i>almdrasata</i>	the school
NCSgFGD	Singular, feminine, genitive, definite common noun	المدرسة	<i>aladrasati</i>	the school
NCDuMNI	Dual, masculine, nominative, indefinite common noun	كتابان	<i>kitabān</i>	two books
NCDuMAI	Dual, masculine, accusative, indefinite common noun	كتابين	<i>kitabain</i>	two books
NCDuMGI	Dual, masculine, genitive, indefinite common noun	كتابين	<i>kitabain</i>	two books
NCDuMND	Dual, masculine, nominative, definite common noun	الكتابان	<i>alkitabān</i>	the two books
NCDuMAD	Dual, masculine, accusative, definite common noun	الكتابين	<i>alkitabain</i>	the two books
NCDuMGD	Dual, masculine, genitive, definite common noun	الكتابين	<i>alkitabain</i>	the two books
NCDuFNI	Dual, feminine, nominative, indefinite common noun	مدرستان	<i>mdrasatan</i>	two books
NCDuFAI	Dual, feminine, accusative, indefinite common noun	مدرستين	<i>mdrasatain</i>	two schools
NCDuFGI	Dual, feminine, genitive, indefinite common noun	مدرستين	<i>mdrasatain</i>	two schools
NCDuFND	Dual, feminine, nominative, definite common noun	المدرستان	<i>almdrasatan</i>	the two schools
NCDuFAD	Dual, feminine, accusative, definite common noun	المدرستين	<i>almdrasatain</i>	the two schools
NCDuFGD	Dual, feminine, genitive, definite common noun	المدرستين	<i>almdrasatain</i>	the two schools
NCPIMNI	Plural, masculine, nominative, indefinite common noun	كتب - مسلمون	<i>muslimoon – kutubun</i>	Muslims – books
NCPIMAI	Plural, masculine, accusative, indefinite common noun	كتبا - مسلمين	<i>muslimeen – kutuban</i>	Muslims – books
NCPIMGI	Plural, masculine, genitive, indefinite common noun	كتب - مسلمين	<i>muslimeen – kutubin</i>	Muslims – books
NCPIMND	Plural, masculine, nominative, definite common noun	الكتب - المسلمون	<i>almuslimoon – alkutubu</i>	the Muslims – the books
NCPIMAD	Plural, masculine, accusative, definite common noun	الكتب - المسلمين	<i>aluslimeen – alkutuba</i>	the Muslims – the books
NCPIMGD	Plural, masculine, genitive, definite common noun	الكتب - المسلمين	<i>almuslimeen – alkutubi</i>	the Muslims – the books
NCPIFNI	Plural, feminine, nominative, indefinite common noun	مسلمات - مدارس	<i>mdarisun – muslimaatun</i>	schools – Muslims
NCPIFAI	Plural, feminine, accusative, indefinite common noun	مسلمات - مدارس	<i>mdarisan – muslimaatan</i>	schools – Muslims
NCPIFGI	Plural, feminine, genitive, indefinite, common noun	مسلمات - مدارس	<i>mdarisin – muslimaatin</i>	schools – Muslims
NCPIFND	Plural, feminine, nominative, definite common noun	المسلمات - المدارس	<i>almdarisu – almuslimaatu</i>	the schools – the Muslims
NCPIFAD	Plural, feminine, accusative, definite common noun	المسلمات - المدارس	<i>almdarisa – almuslimaata</i>	the schools – the Muslims
NCPIFGD	Plural, feminine, genitive, definite common noun	المسلمات - المدارس	<i>almdarisi – almuslimaati</i>	the schools – the Muslims
NP	Proper noun	شيرين - جدة	<i>Jiddah – Shyryn</i>	Jeddah – Shereen
NPrPSg1	First person, singular, neuter, personal pronoun	كتابي - ضربني - أنا	<i>ana- kitaabee – qarabanee</i>	Me – my book – he hit me
NPrPSg2M	Second person, singular, masculine, personal pronoun	كتابك - أنت	<i>anta – kitaabuka</i>	You – your book
NPrPSg2F	Second person, singular, feminine, personal pronoun	كتابك - أنت	<i>anti – kitaabuki</i>	You – your book
NPrPSg3M	Third person, singular, masculine, personal pronoun	هو - كتابه	<i>kitaabahu – huwa</i>	His book – him
NPrPSg3F	Third person, singular, feminine, personal	هي - كتابها	<i>kitaabuhaa – hiya</i>	Her book –

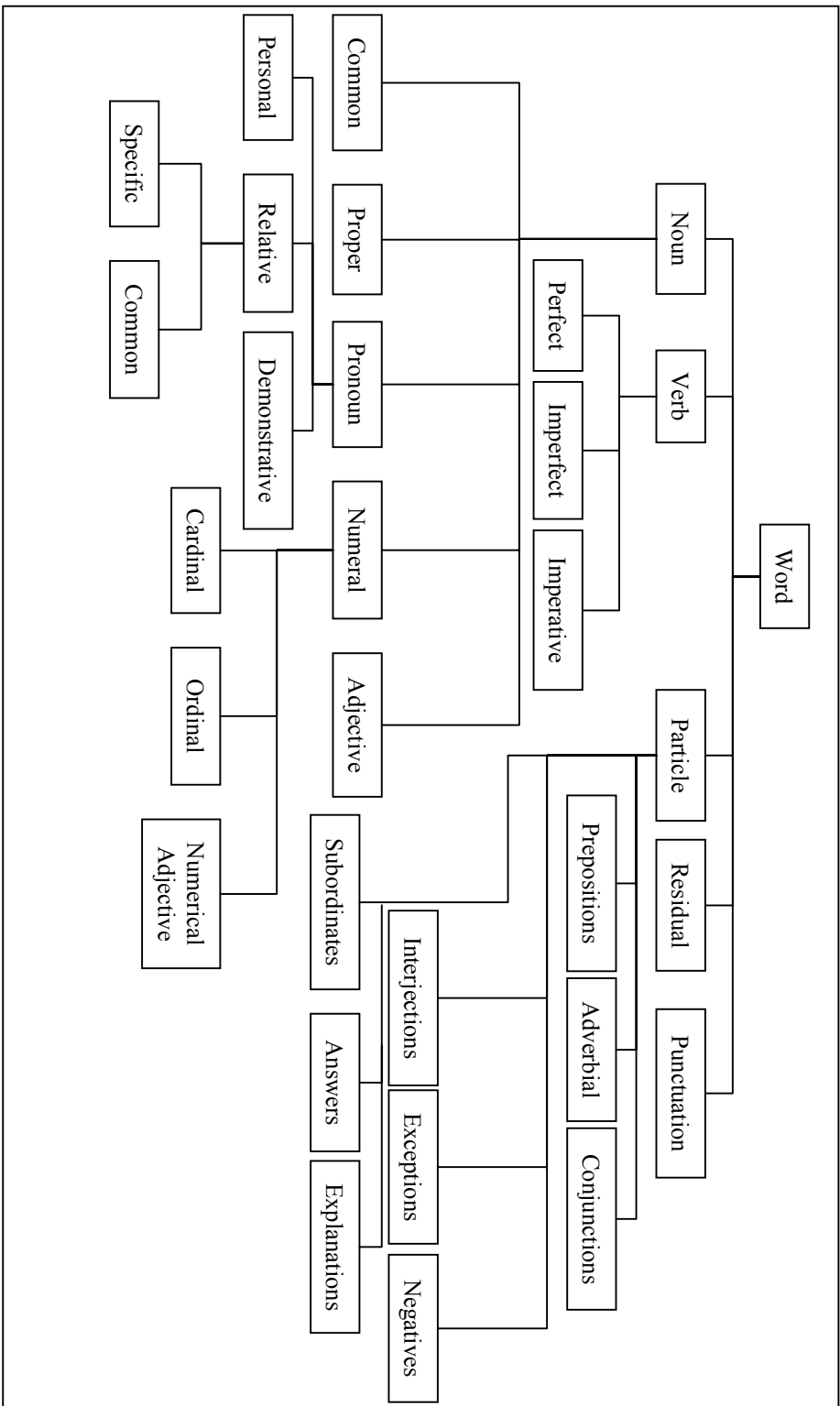
NPrPDu2	pronoun Second person, dual, neuter, personal pronoun	كتابكما-أنتما	<i>antumaa – kitaabakumaa</i>	her You two – your book
NPrPDu3	Third person, dual, neuter, personal pronoun	كتابهما-هما	<i>humaa – kitaabahumaa</i>	Those two – their book
NPrPPI1	First person, plural, neuter, personal pronoun	كتابنا-نحن	<i>nahnu – kitaabunaa</i>	Us – our book
NPrPPI2M	Second person, plural, masculine, personal pronoun	كتابكم-أنتم	<i>antum – kitaabakum</i>	You – your book
NPrPPI2F	Second person, plural, feminine, personal pronoun	كتابكن-أنتن	<i>antunna – kitaabakunna</i>	You – your book
NPrPPI3M	Third person, plural, masculine, personal pronoun	كتابهم-هم	<i>hum – kitaabahum</i>	Them – their book
NPrPPI3F	Third person, plural, feminine, personal pronoun	هن - كتابهن	<i>kitaabahunna – hunna</i>	Their book – them
NPrRSSgM	Singular, masculine, specific, relative pronoun	الذي	<i>allathi</i>	Who
NPrRSSgF	Singular, feminine, specific, relative pronoun	التي	<i>allati</i>	Who
NPrRSDuM	Dual, masculine, specific, relative pronoun	الذين-الذان	<i>alladhani – alladhaini</i>	Who
NPrRSDuF	Dual, feminine, specific, relative pronoun	اللتين-اللتان	<i>allataani – allataini</i>	Who
NPrRSPIM	Plural, masculine, specific, relative pronoun	الذين - اللاتي	<i>allaiy – alladheena</i>	Who
NPrRSPIf	Plural, feminine, specific, relative pronoun	اللاتي - اللاتي	<i>allaaiy - allatee</i>	Who
NPrRC	Common, relative pronoun	مهما- ما -من	<i>men – maa – mahmaa</i>	Who – what
NPrDSgM	Singular, masculine, demonstrative pronoun	ذلك- ذاك - ذا -هذا	<i>hadhaa – dhaa – dhaaka – dhaalika</i>	This – that
NPrDSgF	Singular, feminine, demonstrative pronoun	تلك - ذي - ذه - هذي -هذه تيك-تاك	<i>haadhihi – haadhee – dhih – dhy – tilka – taaka – teeka</i>	This – that
NPrDDuM	Dual, masculine, demonstrative pronoun	- هذين - ذاك - ذان -هذان ذينك-ذين	<i>haadhani – dhaani – dhaanika – hadhaini – dhaini</i>	This – that
NPrDDuF	Dual, feminine, demonstrative pronoun	تين - هتين - تانك - تان -هتان تينك-	<i>haatani – taani-taanika – haataini</i>	This – that
NPrDPI	Plural, neutral, demonstrative pronoun	أولئك - أولاء - أولى - هؤلاء أولئك- أولالك -	<i>haaolaai – olaa-olaaiika- olaalika – olaaka</i>	Those
NNuCaSgM	Singular, masculine, cardinal number	أربع	<i>arba'</i>	Four
NNuCaSgF	Singular, feminine, cardinal number	أربعة	<i>arba'a</i>	Four
NNuOrSgM	Singular, masculine, ordinal number	رابع	<i>raabi'</i>	Fourth
NNuOrSgF	Singular, feminine, ordinal number	رابعة	<i>raabia</i>	Fourth
NNuNaSgM	Singular, masculine, numerical adjective	رباعي	<i>rubaa'y</i>	Of four
NNuNaSgF	Singular, feminine, numerical adjective	رباعية	<i>rubaa'iya</i>	Of four
NACSGMNI	Singular, masculine, nominative, indefinite adjective	سعيد	<i>sa'ydu</i>	happy
NACSGMAI	Singular, masculine, accusative, indefinite adjective	سعيدا	<i>sa'ydan</i>	happy
NACSGMGI	Singular, masculine, genitive, indefinite adjective	سعيد	<i>sa'ydin</i>	happy
NACSGMND	Singular, masculine, nominative, definite adjective	السعيد	<i>alsa'ydu</i>	the happy
NACSGMAD	Singular, masculine, accusative, definite adjective	السعيد	<i>alsa'yda</i>	the happy
NACSGMGD	Singular, masculine, genitive, definite adjective	السعيد	<i>alsa'ydi</i>	the happy
NACSGFNI	Singular, feminine, nominative, indefinite adjective	سعيدة	<i>sa'ydatun</i>	happy

NACSGFAI	Singular, feminine, accusative, indefinite adjective	سعيدتاً	<i>sa'ydatan</i>	happy
NACSGFGI	Singular, feminine, genitive, indefinite adjective	سعيدة	<i>sa'ydatin</i>	happy
NACSGFND	Singular, feminine, nominative, definite adjective	السعيدة	<i>alsa'ydatu</i>	the happy
NACSGFAD	Singular, feminine, accusative, definite adjective	السعيدة	<i>alsa'ydata</i>	the happy
NACSGFGD	Singular, feminine, genitive, definite adjective	السعيدة	<i>alsa'ydati</i>	the happy
NACDuMNI	Dual, masculine, nominative, indefinite adjective	سعيدان	<i>sa'ydan</i>	two happy
NACDuMAI	Dual, masculine, accusative, indefinite adjective	سعيدين	<i>sa'ydain</i>	two happy
NACDuMGI	Dual, masculine, genitive, indefinite adjective	سعيدين	<i>sa'ydain</i>	two happy
NACDuMND	Dual, masculine, nominative, definite adjective	السعيدان	<i>alkitaban</i>	the two happy
NACDuMAD	Dual, masculine, accusative, definite adjective	السعيدين	<i>alsa'ydain</i>	the two happy
NACDuMGD	Dual, masculine, genitive, definite adjective	السعيدين	<i>alsa'ydain</i>	the two happy
NACDuFNI	Dual, feminine, nominative, indefinite adjective	سعيدتان	<i>sa'ydatan</i>	two happy
NACDuFAI	Dual, feminine, accusative, indefinite adjective	سعيدتين	<i>sa'ydatain</i>	two happy
NACDuFGI	Dual, feminine, genitive, indefinite adjective	سعيدتين	<i>sa'ydatain</i>	two happy
NACDuFND	Dual, feminine, nominative, definite adjective	السعيدتان	<i>alsa'ydatan</i>	the two happy
NACDuFAD	Dual, feminine, accusative, definite adjective	السعيدتين	<i>alsa'ydatain</i>	the two happy
NACDuFGD	Dual, feminine, genitive, definite adjective	السعيدتين	<i>alsa'ydatain</i>	the two happy
NACPIMNI	Plural, masculine, nominative, indefinite adjective	سعيدون	<i>sa'ydoon</i>	happy
NACPIMAI	Plural, masculine, accusative, indefinite adjective	سعيدين	<i>sa'ydeen</i>	happy
NACPIMGI	Plural, masculine, genitive, indefinite adjective	سعيدين	<i>sa'ydeen</i>	happy
NACPIMND	Plural, masculine, nominative, definite adjective	السعيدون	<i>alsa'ydoon</i>	the happy
NACPIMAD	Plural, masculine, accusative, definite adjective	السعيدين	<i>alsa'ydeen</i>	the happy
NACPIMGD	Plural, masculine, genitive, definite adjective	السعيدين	<i>alsa'ydeen</i>	the happy
NACPIFNI	Plural, feminine, nominative, indefinite adjective	سعيدات	<i>sa'ydaatun</i>	happy
NACPIFAI	Plural, feminine, accusative, indefinite adjective	سعيداتاً	<i>sa'ydaatan</i>	happy
NACPIFGI	Plural, feminine, genitive, indefinite adjective	سعيدات	<i>sa'ydaatin</i>	happy
NACPIFND	Plural, feminine, nominative, definite adjective	السعيدات	<i>alsa'ydaatu</i>	the happy
NACPIFAD	Plural, feminine, accusative, definite adjective	السعيدات	<i>alsa'ydaata</i>	the happy
NACPIFGD	Plural, feminine, genitive, definite adjective	السعيدات	<i>alsa'ydaati</i>	the happy
VPSg1	First person, singular, neuter, perfect verb	كسرت	<i>kasartu</i>	I broke
VPSg2M	Second person, singular, masculine, perfect verb	كسرت	<i>kasarta</i>	You broke
VPSg2F	Second person, singular, feminine, perfect verb	كسرت	<i>kasarti</i>	You broke
VPSg3M	Third person, singular, masculine, perfect verb	كسر	<i>kasara</i>	He broke
VPSg3F	Third person, singular, feminine, perfect verb	كسرت	<i>kasarat</i>	She broke
VPDu2	Second person, dual, neuter, perfect verb	كسرتما	<i>kasartumaa</i>	You (two) broke

VPDu3M	Third person, dual, masculine, perfect verb	كسرا	<i>kasaraa</i>	They (two) broke
VPDu3F	Third person, dual, feminine, perfect verb	كسرتا	<i>kasarataa</i>	They (two) broke
VPP11	First person, plural, neuter, perfect verb	كسرنا	<i>kasarnaa</i>	We broke
VPP12M	Second person, plural, masculine, perfect verb	كسرتم	<i>kasartum</i>	You broke
VPP12F	Second person, plural, feminine, perfect verb	كسرتن	<i>kasartunna</i>	You broke
VPP13M	Third person, plural, masculine, perfect verb	كسروا	<i>kasaroo</i>	They broke
VPP13F	Third person, plural, feminine, perfect verb	كسرن	<i>kasarna</i>	They broke
VISg1I	First person, singular, neuter, indicative, imperfect verb	أكسر	<i>aksiru</i>	I break
VISg1S	First person, singular, neuter, subjunctive, imperfect verb	أكسر	<i>aksira</i>	I break
VISg1J	First person, singular, neuter, jussive, imperfect verb	أكسر	<i>aksir</i>	I break
VISg2MI	Second person, singular, masculine, indicative, imperfect verb	تكسر	<i>taksiru</i>	You break
VISg2MS	Second person, singular, masculine, subjunctive, imperfect verb	تكسر	<i>taksira</i>	You break
VISg2MJ	Second person, singular, masculine, jussive, imperfect verb	تكسر	<i>taksir</i>	You break
VISg2FI	Second person, singular, feminine, indicative, imperfect verb	تكسرين	<i>taksiryana</i>	You break
VISg2FS	Second person, singular, feminine, subjunctive, imperfect verb	تكسري	<i>taksiry</i>	You break
VISg2FJ	Second person, singular, feminine, jussive, imperfect verb	تكسري	<i>taksiry</i>	You break
VISg3MI	Third person, singular, masculine, indicative, imperfect verb	يكسر	<i>yaksiru</i>	He breaks
VISg3MS	Third person, singular, masculine, subjunctive, imperfect verb	يكسر	<i>yaksira</i>	He breaks
VISg3MJ	Third person, singular, masculine, jussive, imperfect verb	يكسر	<i>yaksir</i>	He breaks
VISg3FI	Third person, singular, feminine, indicative, imperfect verb	تكسر	<i>taksiru</i>	She breaks
VISg3FS	Third person, singular, feminine, subjunctive, imperfect verb	تكسر	<i>taksira</i>	She breaks
VISg3FJ	Third person, singular, feminine, jussive, imperfect verb	تكسر	<i>taksir</i>	She breaks
VIDu2I	Second person, dual, neuter, indicative, imperfect verb	تكسران	<i>taksiraani</i>	You break
VIDu2S	Second person, dual, neuter, subjunctive, imperfect verb	تكسرا	<i>taksiraa</i>	You break
VIDu2J	Second person, dual, neuter, jussive, imperfect verb	تكسرا	<i>taksiraa</i>	You break
VIDu3MI	Third person, dual, masculine, indicative, imperfect verb	يكسران	<i>yaksiraani</i>	They break
VIDu3MS	Third person, dual, masculine, subjunctive, imperfect verb	يكسرا	<i>yaksiraa</i>	They break
VIDu3MJ	Third person, dual, masculine, jussive, imperfect verb	يكسرا	<i>yaksiraa</i>	They break
VIDu3FI	Third person, dual, feminine, indicative, imperfect verb	يكسران	<i>yaksiraan</i>	They break
VIDu3FS	Third person, dual, feminine, subjunctive, imperfect verb	يكسرا	<i>yaksiraa</i>	They break
VIDu3FJ	Third person, dual, feminine, jussive, imperfect verb	يكسرا	<i>yaksiraa</i>	They break
VIP11I	First person, plural, neuter, indicative, imperfect verb	نكسر	<i>naksiru</i>	We break
VIP11S	First person, plural, neuter, subjunctive, imperfect verb	نكسر	<i>naksira</i>	We break
VIP11J	First person, plural, neuter, jussive, imperfect verb	نكسر	<i>naksir</i>	We break
VIP12MI	Second person, plural, masculine, indicative, imperfect verb	تكسرون	<i>taksiroon</i>	You break

VIP12MS	Second person, plural, masculine, subjunctive, imperfect verb	تكسروا	<i>taksiroo</i>	You break
VIP12MJ	Second person, plural, masculine, jussive, imperfect verb	تكسروا	<i>taksiroo</i>	You break
VIP12FI	Second person, plural, feminine, indicative, imperfect verb	تكسرن	<i>taksirna</i>	You break
VIP12FS	Second person, plural, feminine, subjunctive, imperfect verb	تكسرن	<i>taksirna</i>	You break
VIP12FJ	Second person, plural, feminine, jussive, imperfect verb	تكسرن	<i>taksirna</i>	You break
VIP13MI	Third person, plural, masculine, indicative, imperfect verb	يكسرون	<i>yaksiroon</i>	They break
VIP13MS	Third person, plural, masculine, subjunctive, imperfect verb	يكسروا	<i>yaksiroo</i>	They break
VIP13MJ	Third person, plural, masculine, jussive, imperfect verb	يكسروا	<i>yaksiroo</i>	They break
VIP13FI	Third person, plural, feminine, indicative, imperfect verb	يكسرن	<i>yaksirna</i>	They break
VIP13FS	Third person, plural, feminine, subjunctive, imperfect verb	يكسرن	<i>yaksirna</i>	They break
VIP13FJ	Third person, plural, feminine, jussive, imperfect verb	يكسرن	<i>yaksirna</i>	They break
VIvSg2M	Second person, singular, masculine, imperative verb	أكسر	<i>aksir</i>	Break!
VIvSg2F	Second person, singular, feminine, imperative verb	أكسري	<i>aksiry</i>	Break!
VIvDu2	Second person, dual, neuter, imperative verb	أكسرا	<i>aksiraa</i>	Break!
VIvPl2M	Second person, plural, masculine, imperative verb	أكسروا	<i>aksroo</i>	Break!
VIvPl2F	Second person, plural, feminine, imperative verb	أكسرن	<i>aksirna</i>	Break!
PPr	Prepositions	في- مع - من - ل -ك	<i>ka - li - min - ma'a - fy</i>	As - for - from - with - in
PA	Adverbial particles	لن- سوف - ثم -إذا	<i>idhaa - thumma - swf - ln</i>	And then - then - shall - won't
PC	Conjunctions	و- حتى -ف	<i>f - hta - w</i>	So - so - and
PI	Interjections	أيتها-يا	<i>ya - aytha</i>	You
PE	Exceptions	سوى-غير	<i>ghyr - swa</i>	Except
PN	Negatives	لم- لا	<i>la - lm</i>	Not
PW	Answers	لا-أجل	<i>la - ajl</i>	No - yes
PX	Explanations	أي	<i>ay</i>	That is
PS	Subordinates	لو-ما	<i>ma - lw</i>	If
RF	Residual, foreign	روجور	<i>rwjwr</i>	Roger
RM	Residual, mathematical	÷	/	/
RN	Residual, number	3	3	3
RD	Residual, day of the week	الاثنين	<i>alithnyn</i>	Monday
Rmy	Residual, month of the year	محرم	<i>mhrm</i>	muharram
RA	Residual, abbreviation	وأس	<i>was</i>	e.g.
RO	Residual, other	آل	<i>Aal</i>	
PU	Punctuation	؟	?	?

The tagset hierarchy



This paragraph is an extract from Al-Jazirah newspaper dated 1/1/1998. Note that Arabic is written from right to left, hence the change in paragraph direction.

بعث_VPSg3M خادم_NCSgMNI الحرمين_NCDuMAD الشريفين_NCDuMGD الملك_NCSgMND فهد_NP بن_
NCSgMNI عبد_NCSgMAI العزيز_NCSgMAD آل_R سعود_NP برفقة_NCSgFNI تهنئة_NCSgFGI الى_PPr
NCSgFGI الرئيس_NCSgMGD الكسندر_RF كواسنيفيسكي_RF رئيس_NCSgMNI جمهورية_NCSgFGI
بولندا_RF بمناسبة_NCSgFGI اليوم_NCSgMAD الوطني_NCSgMND لبلادهِ_NPrPSg3M NCP1FGI_NPr
وأعرب_VPSg3M الملك_NCSgMND المفدى_NCSgMAD باسمهِ_NPrPSg3M NCSgMGI_NPr باسم_
PC_PPr_NCSgM شعب_NCSgMGI وحكومة_NCSgFGI PC_NCSgFGI المملكة_NCSgFGD العربية_NCSgFGD السعودية_
NCSgFGD عن_PPr اخلص_NCSgFNI التهاني_NCP1MND متمنياً_NCSgMAI لفخامته_
NCSgMNI دوام_PPr NCSgFGD الصحة_NCSgFGD والسعادة_NCSgFGD PC_NCSgFGI ولشعب_
PC_PPr_NCSgMGI بولندا_RF الصديق_NCSgMND الازدهار_NCSgMND الدائم_NCSgMN_PU.

Annex 4: The PADT tag set

Aliwy, A.H. 2013. Arabic Morphosyntactic Raw Text Part of Speech Tagging System , PhD thesis, Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, pp 35

Tag	Remark	Tag	Remark	Tag	Remark
VI	imperfect verb	Y	Abbreviation	C	Conjunction
VP	perfect verb	S	Pronoun	P	Preposition
VC	imperative verb	SD	demonstrative pronoun	I	Interjection
N	Noun	F	particle	G	Graphical symbol
A	Adjective	FI	interrogative particle	Q	Number
D	Adverb	FN	negative particle	--	Isolated definite article
Z	Proper noun				

Part-of-Speech for the PADT tagset

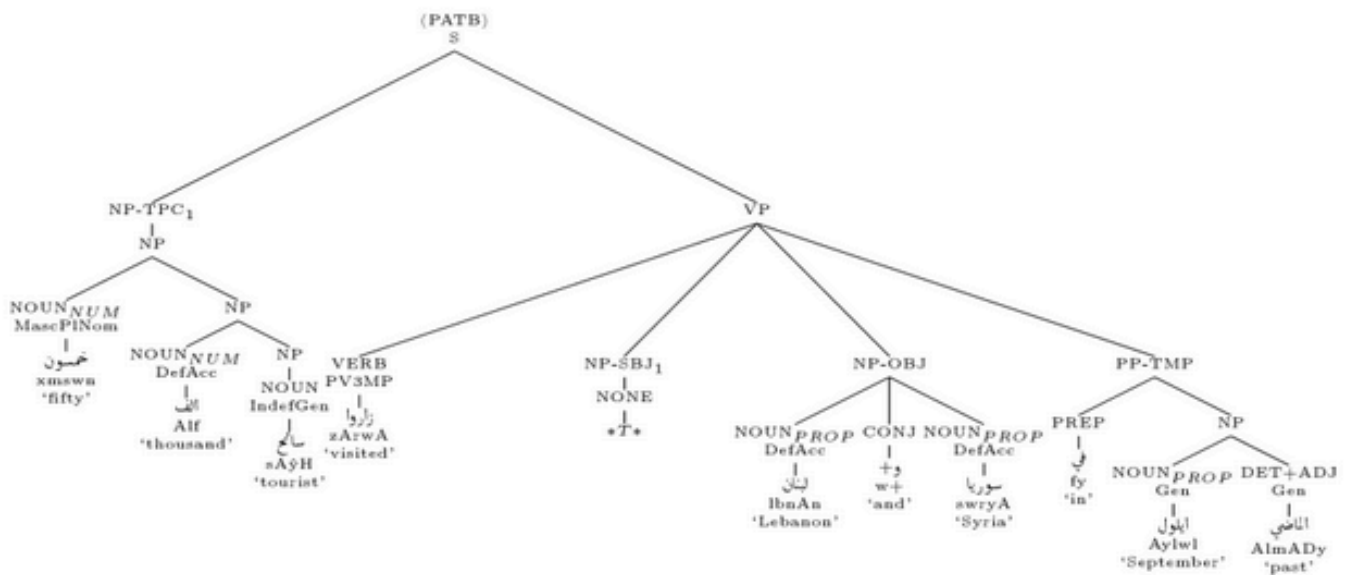
Mood	Indicative	Subjunctive	Jussive	D (ambiguous)
Voice	Active	Passive		
Person	1 speaker	2 addressee	3 others	
Gender	Masculine	Masculine		
Number	Singular	Dual	Plural	

The PADT features

Annex 5: The Penn Arabic Treebank (PATB)

HABASH, N.Y. 2010. Introduction to Arabic Natural Language Processing . Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, pp 105.

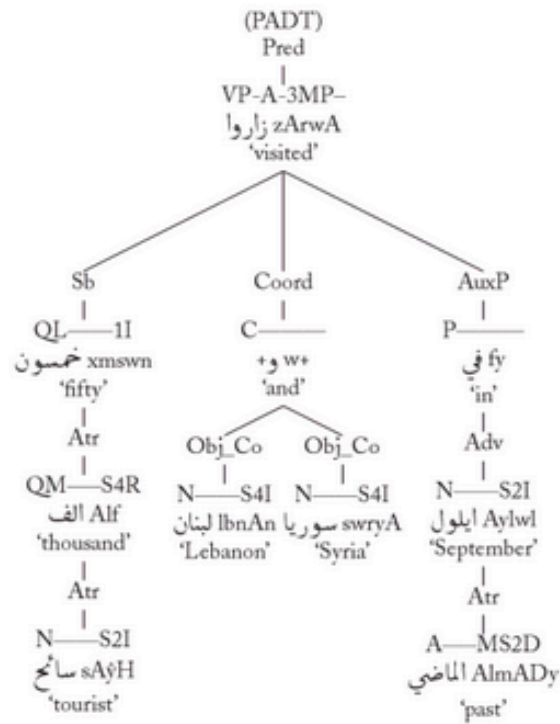
The phrase structure representation in the Penn Arabic Treebank (PATB) for the sentence
 خمسون الف سائح زاروا لبنان وسوريا في ايلول الماضي
xmswn Alf sAyH zArwA lbnAn wswryA fy Aylw AlmADy '50 thousand tourists visited Lebanon and Syria last September.'



Annex 6: The Prague Arabic Dependency Treebank (PADT)

HABASH, N.Y. 2010. Introduction to Arabic Natural Language Processing . Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, pp 107.

The dependency representation in PADT for the sentence
 خمسون ألف سائح زاروا لبنان وسوريا في ايلول الماضي
xmswn Alf sAýH zArwA lbnAn wswryA fy Aylwl AlmADy '50 thousand tourists visited Lebanon and Syria last September.'



Annex 7: The Columbia Arabic Treebank (CATiB)

HABASH, N.Y. 2010. Introduction to Arabic Natural Language Processing . Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, pp 109.

The dependency representation in CATiB for the sentence
 خمسون الف سائح زاروا لبنان وسوريا في ايلول الماضي
xmswn Alf sAÿH zArwA lbnAn wswryA fy Aylwl AlmADy '50 thousand tourists visited Lebanon and Syria last September.'

