Clitics in Arabic Language: A Statistical Study *

Fahad Alotaiby[a], Salah Foda[a], and Ibrahim Alkharashi[b]

[a]Department of Electrical Engineering, King Saud University,
P.O. Box 800, Riyadh 11421, Saudi Arabia
{falotaiby, sfoda}@hotmail.com

[b]Computer and Electrical Research Institute, King Abdulaziz City for Science and Technology,
P.O. Box 6086, Riyadh 11442, Saudi Arabia
kharashi@kacst.edu.sa

Abstract. Clitics in Arabic language can be attached to a stem or to each other without orthographic marks such as an apostrophe. In this paper we present a statistical study of clitics and its effect in Arabic language. We tokenize large Arabic text using white-spaces and an automatic clitics tokenizer (AMIRA 2.0) and compare the unique-word count in both cases with English language. We also show the resulted distribution of clitics in Arabic and examine the performance of the used tokenizer. Using a 600 million words Arabic corpus, we report that the corresponding lexicon size could be reduced by 24.54% when applying clitics tokenization.

Keywords: Arabic, clitics, statistics.

1 Introduction

Arabic language is a rich and complex language. In Arabic, clitics are attached to a stem or to each other without any orthographic marks (i.e. an apostrophe). Number of clitics in Arabic is limited. When concatenated, clitics can generate a chain of up to four clitics before the stem (proclitics) and three clitics after the stem (enclitics). This may reduce the total number of white-spaces delimited words in a script but increases the size of the corresponding lexicon.

Alotaiby *et al.* (2009) showed that the total number of unique words was about 2.2 million in a 600 million words Arabic corpus while in an equivalent English corpus it was 1.26 million. This implies the need of an Arabic lexicon of a size equivalent to 1.76 times the size of an English lexicon used to cover a similar broad linguistic content. A justification of this ratio was the heavy existence of clitics in Arabic language. A good approach to investigate this is by extracting clitics out of stems and performing comparative statistical analysis.

1.1 Arabic Language

Arabic language is a Semitic language that is spoken by more than 280 million. Classical Arabic writing system was originally consonantal and written from right to left. Every letter in the 28 Arabic alphabets represents a single consonant. To overcome the problem of different pronunciations of consonants in Arabic text, graphical signs known as diacritics were invented in the seventh century. Currently in the Modern Standard Arabic (MSA), diacritics are omitted from written text almost all the time. As a result, this omission increases the number homographs

* This work has been supported by a direct grant from His Excellency the Rector of King Saud University Prof. Abdullah Bin Abdulrahman Al-Othman, and by a grant from the Research Center, College of Engineering, King Saud University.

(words with the same writing form). However, Arab readers normally differentiate between homographs by the context of the script.

Moreover, Arabic is a morphologically complex language. An Arabic word may be constructed out of a stem plus affixes and clitics. Furthermore, some parts of the stem may be deleted or modified when appending a clitic to it according to specific orthographical rules. As a final point, different orthographic conventions exist across the Arab world (Buckwalter, 2004*a*). As a result of omitting diacritics, complex morphology and peculiar orthographical rules, processing Arabic text is a difficult task whether performed using statistical or rule-based methods.

1.2 Clitics in Arabic

A clitic is a linguistic unit that is pronounced and written like an affix but it is grammatically independent. Linguistically speaking, if one can parse an Arabic linguistic unit attached to a stem it should be considered as a clitic. This covers most of the clitics except the definite article (الـ) (*Al*) (*the*). It is worth to mention that the transliteration used in this work is based on (Buckwalter, 2004*b*). In the worst case, there could be four concatenated proclitics and three enclitics attached to stems like (أفبالباطل) (*> f b Al bATl*) (*? then by the false*) (*then by the false?*) and (وهبتنيه) (*whb t ny h*) (*gave you me it*) (*you gave it to me*) respectively. Figure 1 shows a model for constructing main tokens in Arabic, where the number over the link represents the possible number of iterations keeping in mind the rules of ordering clitics. In Arabic an enclitic may immediately come after a proclitic as in (بها) (*b ha*) (*by it*) (*by it*).

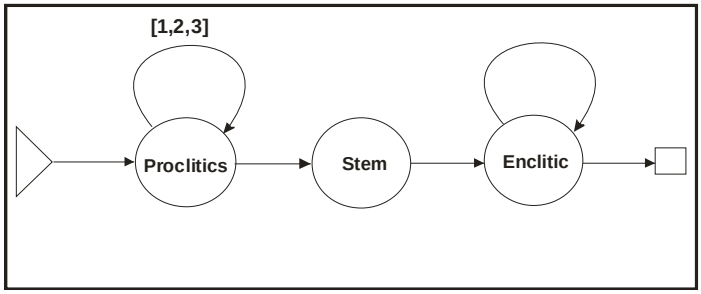

Figure 1: A model for constructing tokens in Arabic.

Proclitics and enclitics processed in this work are listed in Table 1, where enclitics are marked by "+" at the beginning and proclitics are marked by "#" at the end. All the clitics mentioned in Table 1 are extracted from (Maamouri *et al.* 2007) except the definite article. Besides these clitics, there are some other clitics that are rarely used in the MSA like the preposition (تاء القسم) (*t*) (*by*) (*swear by*) used as in (تالله) (*tAllAh*) (*by Allah*) (*swear by Allah*) and the interrogative particle (أ) (>) used as in (أذهبت للمدرسة؟) (*> *hb t*) (*? went you to school*) (*did you go to school?*). In addition, some complex structures are hardly ever used in the MSA like the token (أَوَستعطونيها؟) (>wstETwnyhA) (*? and will give you me it*) (*and will you give it to me?*).

2 Tokenization

The early step of processing any text is to divide the text into proper linguistic units (Manning, 1999). The resulting unit is called token and the process is called tokenization. The naive scheme of tokenization is to divide the text between white spaces or punctuation marks into tokens. In Arabic, this could produce a grammatically complete sentence like (وسيكتبونها) (*wsyktbwnhA*) (*and will write they it*) (*and they will write it*).

Throughout this work, white-space delimited word will be termed *main token* and divided main token will be called *sub-token*.

Table 1: List of processed clitics in Arabic.

Clitics	Transliteration	Category
ال	*Al#*	Definite Article
و	*w#*	Conjunction, coordinating
ف	*f#*	Conjunction, subordinating
ل	*l#*	Preposition
ب	*b#*	Preposition
ك	*k#*	Preposition
س	*s#*	Future verbal particle
ي	*+y*	POSS_PRON_1S/ PRON_1S
ا	*+A*	PRON_1P
ني	*+ny*	IVSUFF_DO/PRON_1S/PVSUFF_DO
ك	*+k*	POSS_PRON_2MS/ PRON_2MS
كما	*+kmA*	POSS_PRON_2D/ PRON_2D
كم	*+km*	POSS_PRON_2MP/ PRON_2MP
كن	*+kn*	POSS_PRON_2FP/ PRON_2FP
ه	*+h*	POSS_PRON_3MS/ PRON_3MS
ها	*+ha*	POSS_PRON_3FS/ PRON_3FS
هما	*+hmA*	POSS_PRON_3D/ PRON_3D
هن	*+hn*	POSS_PRON_3FP/ PRON_3FP
هم	*+hm*	POSS_PRON_3MP/ PRON_3MP
نا	*+nA*	POSS_PRON_1P/ PRON_1P

2.1 Tokenization Systems

Several tools have been developed to perform clitics tokenization. Each one of these tools applies different approach and has different assumptions. For example, MADA+TOKAN (Habash, 2009) utilizes the morphological information and disambiguation analysis that have been produced in a previous step to generate tokenized text with different possible schemes. In the work of Attia (2007), a rule-based tokenizer was described. It utilized white-space normalization and token filter in two steps to generate tokens. AMIRA (Diab *et al.*, 2004) and its successor AMIRA 2.0 (Diab, 2009) are collections of tools for processing Arabic language based on supervised learning..

2.2 AMIRA 2.0

AMIRA 2.0 is a toolkit for processing Arabic text. It includes a tokenizer, part of speech tagger and a phrase chunker. It employs support vector machine using YAMCHA toolkit to apply supervised learning with no explicit knowledge of deep morphology (Diab, 2009). In AMIRA 2.0, clitic tokenization is performed as characterized by Linguistic Data Consortium (LDC). The defined scheme of tokenization according to the LDC is separating definite articles, conjunctions, attached prepositions, future verbal particle and pronouns from the stem of the word. However, sometimes a preposition could be directly attached to a pronoun. In this work, AMIRA 2.0 was used to perform tokenization process due to the availability, ease of use and reported F score

measure of 99.2%. AMIRA 2.0 has two models for tokenization one of them is much faster but slightly less accurate than the other.

3 Preparing the Data

The corpora used in this work come from English Gigaword (Graff *et al.*, 2007) and Arabic Gigaword (Graff, 2007). They are collections of text data extracted from newswire archives of English and Arabic news sources that have been gathered over several years by the Linguistic Data Consortium (LDC) at the University of Pennsylvania. Text data in the Arabic Gigaword were collected from four newspapers and two press agencies. Text collected from press agencies was typed with less concern since it will be edited before publishing. The Arabic Gigaword corpus contains almost two million documents with nearly 600 million words, while the English Gigaword contains more than seven million documents with three billion words. To be consistent, only 600 million words are randomly extracted from English documents.

A common problem in the Arabic corpus is the omission of white spaces between main tokens that end with graphically non-connecting characters as in (ومكافحةانتشارالاسلحةالنوويةمشيرا) (*and combating the spread of nuclear weapons pointing*) which is a phrase of five connected words. Other problems in this corpus are the inconsistence use of punctuation marks, the presence of odd control characters and existence of many spelling errors. Simple preprocessing has been applied to overcome straight forward problems like splitting the above linked sentence.

4 Statistical Results

Running AMIRA 2.0 to tokenize such a huge corpus is a time consuming task. Therefore, the fast tokenization option was used. Nevertheless, it took about 300 hours of processing on a quad processor PC. The first step after tokenizing the Arabic text was counting every extracted clitic. In addition, frequency of clitics was calculated from a much smaller but manually tokenized corpus which is the LDC's Arabic Treebank part 3 (Maamouri *et al.,* 2007).

Table 2 shows the frequency and percentage of every clitics appeared in the tokenized text ordered in a descending manner in addition to the frequency of clitics appeared in the LDC's Arabic Treebank. Since the definite article and the future verbal particle were not found in the LDC's Arabic Treebank, it is worthless to calculate the overall percentage in the manual case. In general, the frequency of clitics in manual and automatic cases are proportional except the cases of the clitics (*k#*) and (*+kn*). This could be due to the use of the slightly less accurate fast version of AMIRA 2.0. However, false proclitics consisting of the letter (ا) (A) and every other character in the alphabet frequently appeared in the automatically tokenized text using AMIRA 2.0. Among them only the definite article (ال) (Al) is correct. Table 3 shows a list of some false clitics generated from the tokenization process. It is important to mention that the false clitic (ا) (*A#*) is different than the interrogative particle (أ) (*>#*).

The frequency of every token appeared in the Arabic, tokenized Arabic and English text has been counted. The 600,000,000 word Arabic corpus has 2,207,637 unique tokens. After tokenizing the corpus using AMIRA 2.0, the size of the corpus became 848,000,000 and the number of unique tokens became 1,665,899. Simply speaking, the same Arabic corpus has 2.21 million main tokens or 1.67 million sub-tokens. As a result, clitics tokenization reduces the unique number of token by 24.54%.

Table 2: Statistics of Clitics in Arabic Corpus.

Clitic	Transliteration	Frequency of clitics in (Graff *et al.* 2007) using AMIRA 2.0	Percentage (%)	Frequency of clitics in (Maamouri *et al.*, 2007) using manual annotation
ال	*Al#*	103,015,016	57.02	N/A
و	*w#*	31,014,498	17.17	43,716
ل	*l#*	11,470,854	6.35	14,705
ه	*+h*	10,021,855	5.55	12,499
ب	*b#*	8,857,080	4.9	13,049
ها	*+ha*	7,975,714	4.41	10,403
هم	*+hm*	2,264,578	1.25	3,014
ف	*f#*	1,462,196	0.81	2,216
س	*s#*	1,348,962	0.75	N/A
نا	*+nA*	1,144,143	0.63	1,715
ك	*k#*	737,638	0.41	480
هما	*+hmA*	449,528	0.25	606
ي	*+y*	302,246	0.17	457
ك	*+k*	267,685	0.15	254
ني	*+ny*	147,805	0.08	201
كم	*+km*	125,782	0.07	182
هن	*+hn*	35,215	0.02	22
ا	*+A*	21,342	0.01	30
كما	*+kmA*	5,611	0	1
كن	*+kn*	1,057	0	15
Total		180,665,805		103,565

Table 3: A List of some False Clitics and their Counts.

Main token sample	False clitic	count
اكثر	*A#*	528,932
اعمال	*AE#*	156,056
ونحن	*+nHn*	70,219

To get a comprehensible view of tokens statistics, corpora were processed at different sizes starting from 10 to 600,000,000 tokens with the following steps

$$n10^m;\ 1 \leqslant n \leqslant 9,\ 1 \leqslant m \leqslant 8,\ n10^m \leqslant 6 \times 10^8 \quad (1)$$

Figure 2 shows the count of unique tokens in every corpus as the sizes of the corpora grew. As expected, it is clear that clitics tokenization reduces the number of unique tokens in the Arabic corpus. But it is important to keep in mind that the accuracy of the tokenizer is not totally perfect.

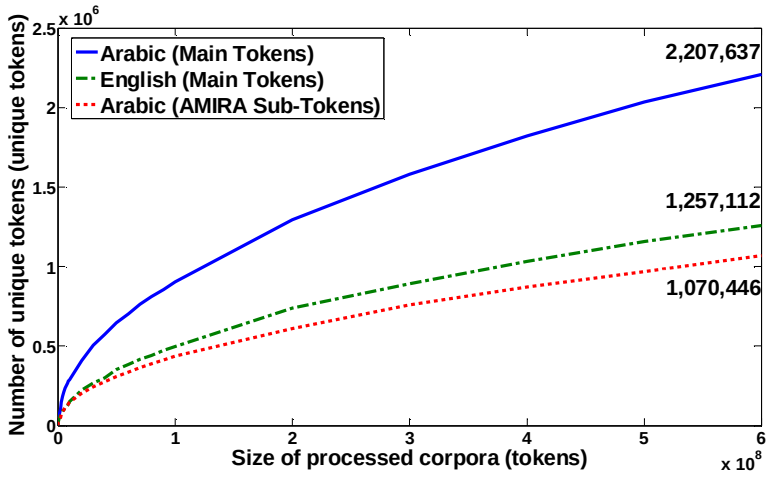

Figure 2: Frequency of unique tokens in every corpus processed at different sizes.

Figure 3 shows the ratio between unique main tokens in Arabic and English languages with average ratio of 1.756. Also, it shows the ratio between unique sub-tokens in Arabic and main token in English with an average ratio of 0.852. However, Arabic has more morphological inflection than English. One would expect Arabic to have more unique word types even after splitting clitics. But it is important to remember that number of homographs in Arabic is much more than it in English which may explain this phenomenon.

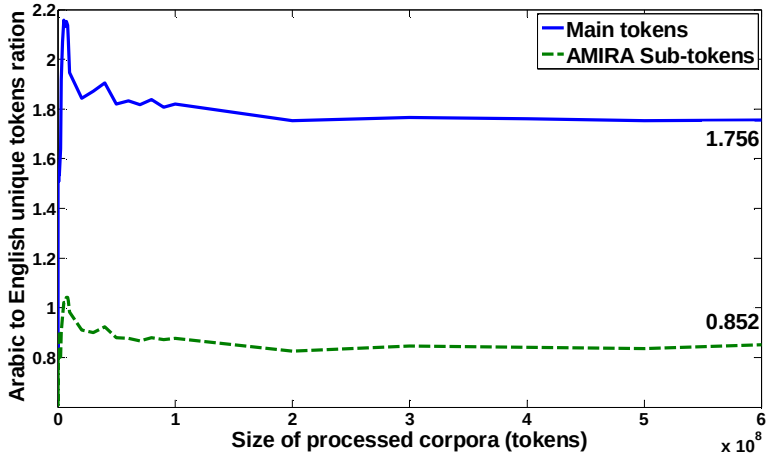

Figure 3: Ratio between Arabic to English unique main tokens and sub-tokens.

5 Conclusion

Clitics tokenization has a great statistical influence in Arabic language. We showed that a reduction of 24.54% in the number of unique tokens has been achieved after applying clitics tokenization on a large Arabic corpus. Also, we showed the distribution of clitics in Arabic and examined the performance and possible frequent mistakes of AMIRA 2.0. In any application that needs an Arabic lexicon such as Automatic Speech Recognition (ASR) systems, it is recommended to use a lexicon of sub-tokens instead of main tokens. This shall reduce the memory need and increase the coverage of the linguistic content of Arabic. Studying the performance of such systems after applying clitics tokenization is a possible future work.

References

Alotaiby, F., I. Alkharashi and S. Foda. 2009. Processing large Arabic text corpora: Preliminary analysis and results. *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, pp. 78-82.

Attia, M. 2007. Arabic tokenization system. *Proceedings of the 2007 Workshop on Computational Approaches To Semitic Languages: Common Issues and Resources*, Prague, Czech Republic. Association for Computational Linguistics, pp. 65-72.

Buckwalter, T. 2004*a*. Issues in Arabic Orthography and Morphology Analysis. *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Geneva, Switzerland.

Buckwalter, T. 2004*b*. *Buckwalter Arabic Morphological Analyzer Version 2.0*, Linguistic Data Consortium (LDC) catalogue number LDC2004L02, ISBN 1-58563-324-0, Philadelphia.

Diab, M. 2009. Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and phrase chunking. *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, pp. 285-288.

Diab, M., K. Hacioglu and D. Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. *5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, Boston, MA.

Graff, D. 2007. *Arabic Gigaword Third Edition*. Linguistic Data Consortium. Philadelphia, USA.

Graff, D., J. Kong, K. Chen and K. Maeda. 2007. *English Gigaword Third Edition*. Linguistic Data Consortium. Philadelphia, USA.

Habash, N., O. Rambow and R. Roth. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, pp. 102-109.

Maamouri, M., A. Bies, S. Kulick, F. Gaddeche, and W. Mekki. 2007. *Arabic Treebank: Part 3(a) v. 2.6*. Linguistic Data Consortium. Philadelphia, USA, Catalog ID: LDC2007E65.

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