Arabic Corpus Enhancement using a New Lexicon/Stemming Algorithm

Ashraf AbdelRaouf[1], Colin A. Higgins[1], Tony Pridmore[1] and Mahmoud I. Khalil[2]
[1]*School of Computer Science, The University of Nottingham, Nottingham, U.K.*
[2]*Faculty of Engineering, Ain Shams University, Cairo, Egypt*

Keywords: Arabic Corpus, Optical Character Recognition, Data Retrieval, Morphological Analysis, Lexicon, Stemming Algorithm.

Abstract: Optical Character Recognition (OCR) is an important technology and has many advantages in storing information for both old and new documents. The Arabic language lacks both the variety of OCR systems and the depth of research relative to Roman scripts. An authoritative corpus is beneficial in the design and construction of any OCR system. Lexicon and stemming tools are essential in enhancing corpus retrieval and performance in an OCR context. A new lexicon/stemming algorithm is presented based on the Viterbi path method which uses a light stemmer approach. Lexicon and stemming lookup is combined to obtain a list of alternatives for uncertain words. This list removes affixes (prefixes or suffices) if there are any; otherwise affixes are added. Finally, every word in the list of alternatives is verified by searching the original corpus. The lexicon/stemming algorithm also assures the continuous updating of the contents of the corpus presented by (AbdelRaouf et al., 2010), which copes with the innovative needs of Arabic OCR research.

1 INTRODUCTION

A corpus is a structured collection of text covering a large number of words from different domains of a given language. The first modern corpus was the Brown Corpus. It was collected and compiled in 1967 (Kučera and Francis, 1967) and contains almost 1 million English words from different disciplines. Currently, the three best known English corpora are: The Corpus of Contemporary American English, containing over 410 million words, created in 1990; the British National Corpus, containing 100 million words, created in 1980 (Corpus, 2007); and the Time Archive, created in 1923 and based on Time magazine. The Time Archive contains more than 275,000 articles with around 100 million words (Time, 2008). All of these continue to be updated.

Stemming is the process of determining the morphological root of a word as well as a tool for data retrieval from a corpus to minimize mismatching errors (Larkey et al., 2002). This is acheived by removing affixes (prefixes, infixes or suffixes) from the word. The word may have many forms while retaining the same meaning, for example, in English "works, working, worker and worked" are all derived from the root word "work". All languages contain nouns and verbs, for example in English the nouns "book, books" are derived from the root word "book" but one is singular and the other plural. Also verbs like "play, played, playing" have the same root word "play" but in different tenses. A sophisticated lexicon need not make all these alternative forms of words explicit, but may be optimized to store only the root words and methods to obtain and report the derived words.

The Arabic language has similar rules to English. Arabic nouns like "كتاب كتابان كتب" are derived from the root word "كتب" but the first is singular, the second is for a pair and the third is plural. Verbs like "لعب يلعب سيلعب" are also derived from the root word "لعب" but in different tenses.

This paper presents an automated way to increase the number of words in an Arabic corpus. It consists of applying a light stemming algorithm to remove all affixes from a word, and then growing the word with affixes to obtain multiple alternative words. These are then either checked against the original corpus or submitted, for review, to a user. The paper is arranged as follows: Section one is an introduction to the paper describing the motivation

AbdelRaouf A., A. Higgins C., Pridmore T. and I. Khalil M. (2013).
Arabic Corpus Enhancement using a New Lexicon/Stemming Algorithm.
In *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods*, pages 435-440
DOI: 10.5220/0004260704350440

for applying the lexicon/stemming algorithm to the corpus, showing Arabic morphology and describing other work that has contributed this strategy. Section two describes the approach that had been followed to solve the problem and the affixes used to enhance the performance of the stemmer. Section three explains the algorithm used to apply the new approach. Section four presents a statistical analysis of the new method. Finally, section five describes the planned future development and uses of the approach and presents some conclusions.

1.1 Motivation

Natural Language Processing (NLP) is the use of computer technologies for the creation, archiving, processing and retrieval of machine processed language data and is a common research topic involving computer science and linguistics (Maynard et al., 2002). Research in the NLP of Arabic are very limited (AbdelRaouf et al., 2010). So, for instance, the Arabic language lacks a robust Arabic corpus. The creation of a well-established Arabic corpus encourages Arabic language research and enhances the development of Arabic OCR applications.

This paper presents a new approach which extends and develops that reported in (AbdelRaouf et al., 2008, AbdelRaouf et al., 2010). An Arabic corpus of 6 million Arabic words containing 282,593 unique words was constructed. In order to check the performance and accuracy of this corpus, a testing dataset of 69,158 words was also created. Upon searching, 89.8% of the testing dataset was found to exist in the corpus. We considered this accuracy very low. To improve this the system was enhanced using a lexicon/stemming algorithm. A combination of stemming and lexicon lookup was used to provide a list of alternatives for the missing words.

We designed our stemmer to avoid two common errors. The first error occurs when the stemmer fails to find the relevant words (words derived from the same root word) and hence fails to increase the corpus accuracy. The second error occurs when the stemmer uses many affixes to create a very long list of alternative words, and hence detects irrelevant words (words not related in meaning to the original word). This also makes it slower.

Our stemmer increases the accuracy of the corpus and simultaneously improves the reporting of relevant words.

1.2 Arabic Language Morphology

The Arabic language depends mainly on the root of a word. The root word can produce either a verb or a noun, for instance “عمل” - a root word – can be a noun as in “عامل معمل” or a verb as in “يعمل تعملون”.

Stemmers, in general, tend to extract the root of the word by removing affixes. English stemmers remove only suffixes whereas Arabic stemmers mainly remove prefixes and suffixes, some of them also remove infixes.

Lexica on the other hand create a list of alternative words that can be produced by that root (Al-Shalabi and Evens, 1998, Jomma et al., 2006).

Arabic words change according to the following variables: (Al-Shalabi and Evens, 1998, Al-Shalabi and Kanaan, 2004)

- *Gender:* Male or female, as in (يعمل تعمل).
- *Tense (verbs only):* Past, present or future, as in (عمل يعمل سيعمل).
- *Number:* Singular, pair or plural, as in (تعمل تعملان تعملون).
- *Person:* First, second or third, as in (عملت عملتن عملن).
- *Imperative verb:* as in (اعمل لاتعمل).
- *Definiteness:* Definite or indefinite, as in (عمل العمل).

The Arabic language, in addition to verbs and nouns, contains prepositions, adverbs, pronouns and so on.

1.3 Related Work

The Arabic language is rich and has a large variety of grammar rules. Research in Arabic linguistics is varied and can be categorized into four main types.

1.3.1 Manually Constructed Dictionaries

A custom Arabic retrieval system is built depending on a list of roots and creates lists of alternative words depending on those roots. This method is limited by the number of roots collected (Al-Kharashi and Evens, 1994).

1.3.2 Morphological Analysis

This is an important topic in natural language processing. It is mainly concerned with roots and stemming identification and is related more to the grammar of the word and its positioning (Al-Shalabi and Evens, 1998, Al-Shalabi and Kanaan, 2004, Jomma et al., 2006).

1.3.3 Statistical Stemmers

These do not depend on the language involved but on the similarity of rules among different languages. A stemmer generator has been developed using a parallel corpus which is a collection of sentence pairs with the same meaning in different languages (Rogati et al., 2003).

1.3.4 Light Stemmers

These depend on affix removal. An affix is defined here as one or more letters added to the root word. This type of stemmer needs less knowledge of Arabic grammar (Aljlayl and Frieder, 2002, Larkey et al., 2002).

2 APPROACH

Lexicon and stemming tools are used in enhancing corpus retrieval and performance in an OCR context. This section presents a lexicon/stemming algorithm based on the use of a light stemmer. Lexicon and stemming lookup is combined to obtain a list of alternatives for uncertain words. This list removes affixes (prefixes or suffixes) if there are any, if not, adds affixes to the uncertain word. Finally, it tries to verify every word in the list of alternatives by searching the original corpus. A tool is added to continually improve the corpus by adding new words and justifying them using the lexicon/stemming algorithm.

2.1 Problem Definition

The approach aims to improve the viability of the Multi-Modal Arabic Corpus (MMAC) (AbdelRaouf, Higgins et al. 2010). This approach helps to facilitate an easy and acceptable way to continuously update MMAC with new words. The approach adds new words using the different linguistic information available for these words. For every new word, the approach gives the following information.

- The existence of the new word in MMAC. This means that the word can be added to the corpus without any problems.
- The existence of an alternative word in MMAC. This is less accurate but it strongly suggests the word might be correct.
- Finally, the non-existence of neither the word nor its alternatives in MMAC. As a result, this word must be carefully checked manually.

The approach offers a tool that allows adding a list of new words to MMAC corpus and regenerates MMAC content files automatically.

2.2 Stemmer used

Of the two main types of stemmer the light stemmer was chosen for the following reasons.

- Light stemmers give more accurate results (Aljlayl and Frieder, 2002).
- Morphological stemmers depend on regular language rules. This may be applicable to the English language because of the relatively small number of irregular rules, but is not applicable to the Arabic language because of the large number of irregular rules.
- Morphological stemmers sometimes depend on diacritics to extract the root. Diacritics are not included in this research.
- A light stemmer does not need good knowledge of the language grammar rules.

2.3 The Proposed Light Stemmer Affixes

In the Arabic language, the root may be prefixed (e.g. عمل تعمل), infixed (e.g. عمل عمال) or suffixed (e.g. عمل عملي) by one or more letters. These letters have no meaning if written separately. Most of the light stemmers remove prefixes and suffixes only.

The affixes used are shown in Table 1 (Larkey, Ballesteros et al., 2002). Shalabi (Al-Shalabi and Kanaan, 2004) created a lexicon that included affixes using morphological rules which are not used in other light stemmers but are used here. These affixes where chosen as they proved most useful during testing, however, it was clear that more affixes were needed to improve performance. Table 1 lists the different types of light stemmers. It also shows all affixes used to enable more words to be detected by the corpus.

Table 1: Affixes of different light stemming types.

Stemmer	Prefixes	Suffixes
Shalabi	ي	ا، وا، ت، نا، ن، تما، تن، تم
Light 1	ال، وال، بال، كال، فال	None
Light 2	ال، وال، بال، كال، فال، و	None
Light 3	"	ه، ة
Light 8	"	ها، ان، ات، ون، ين، يه، ية، ه، ة، ي
Light 10	ال، وال، بال، كال، فال،لل ،و	"
Proposed Stemmer	ي، ال، وال، بال، كال، فال، لل،و ،يا ،يال	ا، وا، ت، نا، ن، تن، تم، ها، ان، ات، ون، ين، يه، ه، ي، هم، هن، كم

3 THE PROPOSED LEXICON/ STEMMING ALGORITHM

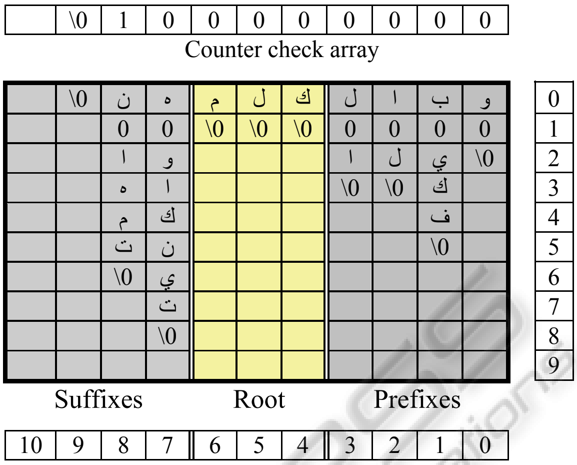
The proposed lexicon/stemmer algorithm creates a list of alternative words instead of the missing word from the corpus. It searches the corpus for the words from the alternative list using a binary search.

3.1 The Alternative List

Generating an alternative list of words is the most important part of the approach. These alternatives are a list of all the words that might be derived from the original word. This part of the algorithm deals with the words from the testing dataset (see section 1.1) which are missing from the corpus. It applies the following steps (see Figure 1):

- It checks the first and last letters of a word and searches for the prefixes and suffixes letters shown in Table 1.
- It creates a two-dimensional array of characters of size 25 × 10. 25 characters are used because this is more than the maximum Arabic word length and 10 characters because this is the maximum number of affixes that can be used with a word.
- It creates a one-dimensional array of 25 characters. This array keeps the path to be followed to generate the alternative word from the other two-dimensional array.
- The two dimensional array is used to generate all possible alternatives, by adding prefixes or suffixes to the word (see Table 1).
- In the case of affix with value ‘0’, it runs the search without this character. For example in Figure 1, the first character is ‘و’. The algorithm will add all the paths once with ‘و’ to the alternative list and once more without it. This means going through all the paths once with the affix and once more without it.
- It starts getting all the possible paths from the two-dimensional array using a Viterbi path algorithm and stops in any column whenever the terminating character ('\0') is found.
- It adds each word created in each possible path to the alternative words list of the missed word.

Figure 1 shows the algorithm used to generate the alternative words list of a sample root word “كلم”. The three letters in the middle are the root word. The letters on the right side show all the possible prefixes. The letters on the left side show all the possible suffixes. The counter check array keeps the location of the path to be followed to generate the alternative words list.



	\0	1	0	0	0	0	0	0	0	0

Counter check array

	\0	ن	ه	م	ل	ك	ل	ا	ب	و	0
		0	0	\0	\0	\0	0	0	0	0	1
		ا	و				ا	ل	ي	\0	2
		ه	ا				\0	\0	ك		3
		م	ك						ف		4
		ت	ن						\0		5
		\0	ي								6
			ت								7
			\0								8
											9

Suffixes Root Prefixes

10	9	8	7	6	5	4	3	2	1	0

Figure 1: The structure of the proposed algorithm.

3.2 Implementing the Proposed Approach

A program was implemented to apply the proposed approach to the corpus data files. A script MS DOS batch script was developed to generate the required overall application. The script uses the proposed algorithm program alongside a second program to generate the new corpus data files.

The input is a list of the new words required to be added to the corpus. The program generates two lists of words, a list of unique words found in the corpus and a list of unique words missed from the corpus.

The list of unique words missed from the corpus is divided into two: a trusted and an untrusted words list. The trusted words list includes the words that have an alternative word in the corpus whereas the untrusted words list includes the words that have not got alternative words in the corpus.

The user is required to edit the two lists to manually check the words that are going to be added to the corpus. After verifying the list of words the application loads the new list of words into the corpus. The application regenerates all the data files of the corpus with the new lists of words.

4 STATISTICAL ANALYSIS OF THE PROPOSED APPROACH

This section presents tests of the proposed approach from two aspects. First, tests for the accuracy of the approach and whether they enhance the accuracy of the corpus are given. Secondly, tests on the effect of the approach to increasing the total number of tokens in the corpus are presented. The tests were applied using the MMAC corpus presented by (AbdelRaouf et al., 2010).

4.1 Accuracy of the Approach

The system was tested against the MMAC corpus data using the testing dataset mentioned in section 1.1. The purpose of this test is to check the accuracy of the approach. It also checks whether the generated list of alternative words includes relevant words or not.

The total number of unique words in the corpus is 282,593. The total number of unique words in the testing dataset is 17,766. The total number of words from the testing dataset found in the corpus is 15,967 with an accuracy of 89.8%. Hence the total number of words from the testing dataset missed from the corpus is 1,799 with an error rate of 10.2%.

The total number of words found in the corpus after applying the approach is 17,431 with accuracy of 98.1%. The total number of words still missing from the corpus is 335 with an error of 1.89%.

The total number of words found in the corpus using the approach only is 1,464 words. These words are either relevant to the missing word which wasn't found in the corpus (1,387 words with an accuracy of 94.7%), or irrelevant to the missing word (77 words with an error of 5.3%).

The previous statistics indicate that the proposed algorithm reaches a high level of accuracy in finding the words (98.1%) with a very minor missing words error factor of 1.89% and also a negligible error in finding irrelevant words of 0.4%.

4.2 Corpus Enhancement

The system was also tested by adding new words to MMAC and seeing the effect of these additions upon the performance of the corpus, with the aim of including more new words on a regular basis.

The MMAC testing dataset was used as a list of new words to be added to the corpus. Section 4.1 shows the number of words that are added to the corpus using the testing dataset. The numbers presented are for both trusted and un-trusted lists.

(AbdelRaouf, Higgins et al. 2010) included new Arabic tokens: Piece of Arabic Word (PAW) and Naked Piece of Arabic Word (NPAW) PAW without dots. MMAC makes innovative use of this new concept of connected segments of Arabic words (PAWs) with and without diacritics marks.

Table 2 shows the number of words that are added to the corpus using the *trusted and untrusted* list of the testing dataset. Table 3 shows the number of words that are added to the corpus using the *trusted* list of the testing dataset. Table 4 shows the number of words that are added to the corpus using the *untrusted* list of the testing dataset.

Tables 2, 3 and 4 show that the proposed approach is working well with the MMAC corpus. They also show that the approach can easily increase the total number of valid words in the MMAC corpus. It is also apparent that the process of adding new lists of words to the corpus is very easy, thus the complicated procedures that were followed to generate the corpus are no longer needed during updates.

Table 2: Number and percentage of tokens using trusted and un-trusted lists.

Description	Before lexicon / stemming	After lexicon / stemming	Percentage increase
Total number of Words	6,000,000	6,069,158	1.15%
Number of Unique words	282,593	284,392	0.64%
Number of Unique Naked words	211,072	212,422	0.64%
Number of Unique PAWs	66,725	67,010	0.43%
Number of Unique Naked PAWs	32,804	32,925	0.37%

Table 3: Total number and percentage of tokens using trusted list.

Description	Before lexicon / stemming	After lexicon / stemming	Percentage increase
Total number of Words	6,000,000	6,069,158	1.15%
Number of Unique words	282,593	284,057	0.52%
Number of Unique Naked words	211,072	212,142	0.51%
Number of Unique PAWs	66,725	66,923	0.30%
Number of Unique Naked PAWs	32,804	32,894	0.27%

Table 4: Total number and percentage of tokens using un-trusted list.

Description	Before lexicon / stemming	After lexicon / stemming	Percentage increase
Total number of Words	6,000,000	6,069,158	1.15%
Number of Unique words	282,593	282,928	0.12%
Number of Unique Naked words	211,072	211,352	0.13%
Number of Unique PAWs	66,725	66,813	0.13%
Number of Unique Naked PAWs	32,804	32,839	0.11%

5 CONCLUSIONS AND FURTHER WORK

The proposed algorithm can be used with any Arabic corpus, not only MMAC. The approach can easily increase the size of the corpus with high accuracy. The existence of this approach encourages researchers and developers to enhance Arabic corpora with new lists of words.

Future work will improve the proposed algorithm to first find the most likely word to the missing one and secondly to recommend the relative frequency of letter pairs bigrams and trigrams frequencies. Finally word pairs and tri-gram frequencies will be investigated. The approach will include infix removal in addition to prefixes and suffices. It will be modified to use intelligent language techniques to simply define the rules of affix addition or removal.

REFERENCES

AbdelRaouf, A., C. Higgins and M. Khalil (2008). A Database for Arabic printed character recognition. The International Conference on Image Analysis and Recognition-ICIAR2008, Póvoa de Varzim, Portugal, *Springer Lecture Notes in Computer Science (LNCS) series.*

AbdelRaouf, A., C. Higgins, T. Pridmore and M. Khalil (2010). "Building a Multi-Modal Arabic Corpus (MMAC)." The International Journal of Document Analysis and Recognition (IJDAR) 13(4): 285-302.

Al-Kharashi, I. A. and M. W. Evens (1994). "Comparing words, stems, and roots as index terms in an Arabic Information Retrieval System." *Journal of the American Society for Information Science* 45(8): 548 - 560.

Al-Shalabi, R. and M. Evens (1998). A Computational Morphology System for Arabic. *Workshop on Computational Approaches to Semitic Languages COLING-ACL98*, Montreal.

Al-Shalabi, R. and G. Kanaan (2004). "Constructing An Automatic Lexicon for Arabic Language." *International Journal of Computing & Information Sciences* 2(2): 114-128.

Aljlayl, M. and O. Frieder (2002). On Arabic search: improving the retrieval effectiveness via a light stemming approach. The Eleventh International Conference on Information and knowledge Management, McLean, Virginia, USA, *Conference on Information and Knowledge Management archive.*

Corpus, T. B. N. (2007). The British National Corpus (XML Edition).

Jomma, H. D., M. A. Ismail and M. I. El-Adawy (2006). An Efficient Arabic Morphology Analysis Algorithm. *The Sixth Conference on Language Engineering, Cairo*, Egypt, The Egyptian Society of Language Engineering.

Kučera, H. and W. N. Francis (1967). "Computational Analysis of Present-Day American English." *International Journal of American Linguistics* 35(1): 71-75.

Larkey, L. S., L. Ballesteros and M. E. Connell (2002). Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. 25th I*nternational Conference on Research and Development in Information Retrieval (SIGIR).*

Maynard, D., V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva and Y. Wilks (2002). "Architectural Elements of Language Engineering Robustness." *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data:* 1-20.

Rogati, M., S. McCarley and Y. Yang (2003). Unsupervised learning of Arabic stemming using a parallel corpus. *The 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan.

Time. (2008). "*Time Archive 1923 to present.*" from http://www.time.com/time/archive/.