Representing Arabic Documents Using Controlled Vocabulary Extracted from Wikipedia

Mohamed I. Eldesouki[*1], Waleed M. Arafa[*2], Kareem Darwish[**3], Mervat H. Gheith[*4]

[*] *Department of Computer and Information Sciences, Institute of Statistical Studies and Research, Cairo University*
5 Dr. Ahmed Zwel Street, Orman, Giza, Egypt

[1]disooqi@ieee.org
[2]waleed_arafa@hotmail.com
[4]mervat_gheith@yahoo.com

[**] *Qatar Computing Research Institute, Qatar Foundation*
Al-Nasr Tower A, 21st Floor, Doha, Qatar

[3]kareem@darwish.org

***Abstract*— One of the key aspects in Information Retrieval is the way to represent documents to be retrieved. Some systems, which use documents' keywords only to represent the documents, might neglect indexing of words that of less meaning. Other systems try to choose the most representative keywords for their documents. The same set of keywords could be used with different levels of analysis to provide different representations for the documents.**

In this work, we used Arabic Wikipedia project as source of controlled vocabulary and use this controlled vocabulary for indexing documents. Our technique is very close to the work of Eldesouki [15]. However, instead of using ids to represent the documents, we use the terms themselves to represent each document.

We examined normalizing the documents before applying our technique. Furthermore, we examined stemming the documents before, after, and while applying our technique. The mean average precision of our technique outperforms light10 stemmer. Although the difference is not statistically significant, our technique shows that many terms produced from just stemming are not significant in representing the documents.

Furthermore, using our technique dramatically decreases the size of the index. Experiments show that our technique reduces about 47.5% of the size of the index build from applying light10 stemmer.

1 INTRODUCTION

One of the key aspects in Information Retrieval is the way to represent documents to be retrieved, a so-called logical view of the document. Some systems use the full set of words to represent documents, whereas others use subset of the words to represent documents in the system. The representing of a document could be viewed as a continuum in which it might shift from a full text representation to a higher level representation specified by a human subject [6].

Some systems, which use documents' keywords only to represent the documents, might neglect indexing of words that of less meaning. Pronouns, prepositions, and conjunctions are the typical examples of such words. Some systems keep a list of such words (stopwords list) to prevent from indexing them. Other systems try to choose the most representative keywords for their documents based on factors such as the frequencies of such keywords, their spread over a single document and others.

The same set of keywords could be used with different levels of analysis to provide different representations for the documents. Using different levels of analysis helps to overcome the problem of matching between two sequences of characters.

Different techniques have been developed to overcome the difficulties for matching process including normalization process, stemming process, morphological analysis process, n-gram for words, using ontologies, etc.

In this work, we investigate representing documents using terms of controlled vocabulary extracted from Arabic Wikipedia project. Using this controlled vocabulary, we use a special n-gram algorithm to identify the entities within the text. We further examined using stemming technique before and after applying our technique. The results are compared to other stemming techniques [14].

The rest of the paper is organized as follows: section 2 presents the previous work; section 3 presents the methods of using the controlled vocabulary to represent the documents; section 4 introduces the terms identification which is the first in our

technique. Section 5 lists the reasons of chosen Wikipedia as a source of controlled vocabulary. Experiments setup, results and discussions are provided in section 6 and 7. The conclusions are derived in section 8.

2 PREVIOUS WORK

Abu El-Khair [1] has examined three examples of Arabic stopwords lists for their effectiveness in information retrieval system.

After morphologically analyzing text, Mansour and his companions [25] assign weights for each terms of a document and then sort the terms in descending order by weight to help selecting them later. The weight of a word depends on three factors; the frequency of occurrence in a document, the count of stem words for that word, and on the spread of the word in the document.

In his work, Mohamed Eldesouki [15] has used the Arabic Wikipedia project to represent each document as a set of ids. Each id represents a single entity in the text of the document. Many forms and variants are encoded within these ids (such as synonyms, acronyms, words with different affixes and different morphological variations). Furthermore, the representation using these ids avoids the problem of polysemy since words with different senses assigned different ids. However, two issues constitute the main obstacles for his approach; the first one is the use of word sense disambiguation technique to disambiguate the right sense of terms that has multiple senses. The other problem is the immature nature of the Arabic Wikipedia project which is yet not contain the sufficient amount of variants and forms to represent all the (or even the majority) of the terms variants

Al-Kharashi [4] tried to use dictionaries of roots and stems, built manually, for each word to be indexed. The roots and stems extracted from a very small collection of text.

Arabic morphological Analyzers have been used to obtain the roots of the words automatically to be indexed. A lot of analyzers exist in that time have been used and evaluated; for example Khoja Morphological Analyzer [19], Tim Buckwalter morphological analyzer 1.0 [24], ALPNET morphological analyzer [7], and Sebawai [10].

A controversial issue at that time was whether to use roots or stems as terms for indexing. Several studies have claimed that roots outperform stems [4], [17], [2] and [9]. However, most of the resent studies found that using stems as index terms outperform roots; [5], [21], [11], [22], [28], [12]. The reason that the former researchers, that found roots better than stems for IR tasks, have done their experiment on small collections of text which is not enough for evaluation.

Using the TREC-2001 Arabic corpus [23], experiments reveal that roots are not suitable because Arabic consists of a few thousands of roots. Analyzing each word to its root would conflate many words of different meaning to the same class. For example, the Arabic words for office, book, Library, writer, and letter have same root.

After TREC Arabic cross-language Information retrieval tracks (CLIR) [16], researchers have directed their research to use stems as index terms. They developed a lot of stemmers to handle Arabic Language in IR context. Many studies have been conducted in stemming techniques; [11], [5], [21], [8], [22], [3], [26], [20], [27], and [13].

3 OUR TECHNIQUE

Our technique is very close to the work of Eldesouki [15]. However, instead of using ids to represent the documents, we use the terms themselves to represent each document. In other words, we use the terms to represent the document if and only if they exist in Arabic Wikipedia as articles' titles. The main idea behind this technique is assuming that noun phrases are more representative than verbs, adjectives and adverbs. And we use Wikipedia as a source of the noun phrases to use as a controlled vocabulary.

We overcome the problem of variants limitation in Arabic Wikipedia by using the best stemming technique which is the light10 stemmer to stem the text; to the best of our knowledge [14].

4 TERMS IDENTIFICATION

The term detection or identification task goes as follows: the document is firstly tokenized. The document is then processed to generate word n-grams. The n-gram generation process differs from the usual way of producing n-gram; See Algorithm in Table I. While the system generates n-grams, it tries to match the n-gram to the variants of each different article's titles that have extracted from Wikipedia. The size of the n-gram, n, is equal to longest variant length. Although, there is small likelihood to produce wrong phrases, the customized method for generating n-gram has the advantage of reducing ambiguity by trying to detect longer phrases first.

Our technique could be used with other text processing technique such as normalization, stemming or even morphological analysis. Our technique could be applied before or after these text processing techniques.

TABLE I
ALGORITHM OF TERMS IDENTIFICATION

Input: *TokensQ* (queue of all document tokens), *synDic* (variants dictionary), *n* (size of n-gram)
Output: list of tokens of identified terms in the document
Algorithm:
1) If TokensQ size = 0, then return;
2) Else If TokensQ size >= n, Choose first *n* tokens from the TokensQ into *nList* (a list of n-gram size).
3) Else, choose ***all*** tokens from the TokensQ into *nList*.
4) Constitute a n-gram by concatenating all the tokens in *nList***.**
5) Try to find the term in the variants dictionary
6) If (variant found in *synDic*)
   a) Consider the tokens of the variant to be indexed
   b) Empty *nList* and dequeue the tokens of the term from the *TokensQ*
   c) Go to step 1.
7) Else (the term has no corresponding in *synDic*)
   a) Then remove one token from the end of *nList*.
   b) Check the size of *nList* after removal
      i) If number of tokens that exist in *nList* = 0, dequeue the last removed token from TokenQ and go to step 1.
      ii) If number of tokens that exist in *nList* > 0, then go to step 4.

5 WIKIPEDIA AS SOURCE OF CONTROLLED VOCABULARY

Wikipedia is a free encyclopedia that maintains topics and subjects that covers many areas of knowledge. Articles of Wikipedia usually describe ideas or define specific terminologies. Wikipedia is not a dictionary; it doesn't contain general words.
We use a controlled vocabulary built from the titles of Wikipedia's article to represent documents. The key idea behind our technique is that instead of using a general dictionary or lexicon to represent document, we use a set of constantly-increasing terminologies to represent the documents.

The continuous growth of the Wikipedia project makes it a good source of a controlled vocabulary. Due to collaboration work of volunteers, the Wikipedia grows constantly and rabidly. This gives it more advantage than other resources which is fixed in size such as Arabic WordNet. The Wikipedia produces a database dump every 15 days. This makes the Wikipedia reflects the reality and makes it up-to-date.

We used Arabic Wikipedia project as source of the controlled vocabulary. The controlled vocabulary has been extracted using two ways. Redirect pages and the anchors' text of interlinks between articles of Arabic Wikipedia.

6 EXPERIMENTS SETUP

The experiments measure the effect of using index terms produced by our technique to improve retrieval effectiveness of the information retrieval system.

As we mentioned earlier, our technique could be used in existence of other text processing steps such as normalization and stemming. We examined normalizing the documents before applying our technique. Furthermore, we examined stemming the documents before, after, and while applying our technique. We choose the light10 stemmer to stem the text since it is the outperforming stemmer [14]. We experiment using a controlled vocabulary extracted from only redirect pages and from both redirect pages and the anchors' text of interlinks between articles. Each experiment is conducted with and without relevance feedback.

The results of our techniques are compared with stemming techniques, since they outperform the other techniques for processing Arabic text [14].

We have used TREC-2001 Arabic corpus for evaluation. TREC-2001 Arabic corpus, also called the AFP_ARB corpus, consists of 383,872 newspaper articles in Arabic from Agence France Presse. This fills up almost a gigabyte in UTF-8 encoding as distributed by the Linguistic Data Consortium. There were 25 and 50 topics used in 2001 and 2002 respectively with relevance judgments, available in Arabic, French, and English, with Title, Description, and Narrative fields. We used the Arabic titles and descriptions as queries of the 75 topics in the experiments.

For all the experiments, we used the Lemur language modeling toolkit [30], which was configured to use Okapi BM-25 term weighting with default parameters and with and without blind relevance feedback (the top 50 terms from the top 10 retrieved documents were used for blind relevance feedback). To observe the effect of alternate indexing terms, mean average precision, MAP, was used as the measure of retrieval effectiveness. To determine if the difference between results was statistically significant, a paired t-test [18] and Wilcoxon sign test [29] have been used with p values less than 0.05 as indication for significance.

As a requirement for Arabic text to be indexed with Lemur toolkit, corpus and topics have been converted to CP1256 encoding. Then a normalization step was performed. The encoding conversion and normalization steps were conducted on both text collection and the topics where queries were extracted. We applied our technique to the topics as required.

In order to be able to compare the retrieval performance with the light stemmers mentioned in [14], the same experiment parameters have been used for current work.

7 Results and Discussion

Table II shows the results of applying our technique after normalizing the documents as well as the results of stemming the documents before, after and while applying our technique.

TABLE II
EXPERIMENTS USING OUR TECHNIQUE (BOTH REFERS TO REDIRECT PAGES AND ANCHORS' TEXT)

	Use Normalized Text		**Stem Text**					
			Before		**Through**		**After**	
	With	**Without**	**With**	**Without**	**With**	**Without**	**With**	**Without**
Redirect only	0.2690	0.2296	0.3791	0.3471	0.3327	0.29	0.3327	0.2848
both	0.3056	0.2470	0.3936	0.3510	0.3521	0.2969	0.3919	0.3496

The experiments are conducted using controlled vocabulary extracted from only redirect pages and from both redirect pages and context of other articles. All experiments are conducted with and without blind relevance feedback.

The results show that using stemming before or after applying technique dramatically increases the performance of the information retrieval system. The table shows that the difference between normalizing text and stemming text before applying our technique is statistically significant where the t-test and sign test values are 0.0002 and 0.00, respectively, with query expansion and when extracting Wikipedia data using both methods.

In the other hand, using both redirect pages and anchors' text dramatically increase the performance of Information Retrieval system over using just the redirect pages.

For using stemming technique, the difference between using stemming technique before applying our technique and after applying our technique is not statistically significant with and without query expansion where t-test is 0.3837 and sign test is 0,3638 when expanding, and t-test is 0.3801 and sign test is 0.1778 when not expanding. We have to note that this result is for using "both" ways of extracting Wikipedia methods. In case of using only redirect pages, the difference between stemming after and stemming before is significant, where the t-test and sign tests are 0.005 and 0.0001, respectively when expanding, and 0.0003 and 0.000, respectively when not expanding.

Table III shows the index sizes for the different experiments. It shows that using both ways for extracting controlled vocabulary always increases the size of the index. Furthermore, stemming the documents after applying our technique gives the smallest index size.

TABLE III
THE SIZES OF INDICES FOR ALL EXPERIMENTS

	Use Normalized Text		**Stem Text**					
			Before		**Through**		**After**	
	With	**Without**	**With**	**Without**	**With**	**Without**	**With**	**Without**
Redirect only	335 MB		471 MB		480 MB		308 MB	
both	424 MB		528 MB		541 MB		382 MB	

Table IV is intended for comparing between our technique, light10 stemmer, and the technique in [15] in terms of performance and index sizes. The table shows that although our technique slightly improves the performance over light10 stemmer, the

different is not statistically significant. However, this could be used as an indication that, when using only stemming, many terms indexed are not important in representing the documents.

Although, our technique adds a burden to the information retrieval system (since it adds another task before or after stemming the text), using our technique dramatically decreases the size of the index by about 47.5%.

TABLE IV
COMPARISON BETWEEN INDEX BUILD FROM THE COLLECTION AFTER APPLYING JUST LIGHT10 STEMMING, OUR TECHNIQUE, AND THE TECHNIQUE IN [15]

	With Query Expansion	Without Query Expansion	Index Size
Technique of [15]	0.3394	0.3813	631 MB
Light10	0.3914	0.3489	727 MB
Our technique (stem first)	0.3936	0.3510	528 MB
Our technique (stem later)	0.3919	0.3496	382 MB

8 CONCLUSIONS

The mean average precision of our technique outperforms light10 stemmer. Although the difference is not statistically significant, our technique shows that many terms produced from just stemming are not significant in representing the documents.

Furthermore, using our technique dramatically decreases the size of the index. Experiments show that our technique reduces about 47.5% of the size of the index build from applying light10 stemmer.

REFERENCES

[1] Abu El-Khair I., 2006, Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study, *International Journal of Computing & Information Sciences*, pages 119-133.

[2] Abu-Salem, H., Al-Omari, M., and Evens, M. Stemming methodologies over individual query words for Arabic information retrieval. *Journal of the American Society for Information Science (JASIS)*, 50 (6), pp. 524-529, 1999.

[3] Al-Ameed k. Hayder, Al-Ketbi O. Shaikha, Al-Kaabi A. Amna, Al-Shebli S. Khadija, Al-Shamsi F. Naila, Al-Nuaimi H. Noura, Al-Muhairi S. Shaikha, Arabic Light Stemmer: A new Enhanced Approach, *The second international conference on innovations technology (IIT'05)*, 2005.

[4] Al-Kharashi, I. and Evens, M. W. Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *Journal of the American Society for Information Science (JASIS)*, 45 (8), pp. 548-560, 1994.

[5] Aljlayl, M., & Frieder, O, On Arabic search: Improving the retrieval effectiveness via light stemming approach. *In Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, Illinois Institute of Technology (pp. 340–347). New York: ACM Press.2002.

[6] Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. 1999. Modern Information Retrieval. AddisonWesley.

[7] Beesley, K. R. Arabic finite-state morphological analysis and generation. *In COLING-96: Proceedings of the 16th international conference on computational linguistics*, vol. 1, pp. 89-94, 1996.

[8] Chen, A., and Gey, F. Building an Arabic stemmer for information retrieval. *In TREC 2002. Gaithersburg: NIST*, pp 631-639, 2002.

[9] Darwish, K., Doermann, D., Jones, R., Oard, D., and Rautiainen, M. *TREC-10 experiments at Maryland: CLIR and video*. In TREC 2001. Gaithersburg: NIST, 2001.

[10] Darwish, K. Building a shallow morphological analyzer in one day. *ACL 2002 Workshop on Computational Approaches to Semitic languages*, July 11, 2002.

[11] Darwish, K. and Oard, D.W. CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval. *In TREC 2002. Gaithersburg: NIST*, pp 703-710, 2002.

[12] Darwish K., Hassan H., and Emam O., Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 25–30, Ann Arbor, June 2005.

[13] El-Beltagy S., Rafea A.. A FRAMEWORK FOR THE RAPID DEVELOPMENT OF LIST BASED DOMAIN SPECIFIC ARABIC STEMMERS, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 2009.

[14] Eldesouki M., Arafa W. and Darwish K. Stemming techniques of Arabic Language: Comparative Study from the Information Retrieval Perspective. *The Egyptian Computer Journal* , Vol. 36 No. 1, June 2009.

[15] Eldesouki M., Arafa W., Darwish K., and Gheith M., Using Wikipedia for Retrieving Arabic Documents, *Proceedings of Arabic Language Technology International Conference*, October 2011.

[16] Gey, F. C. and Oard, D. W. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French, or Arabic queries. *In TREC 2001. Gaithersburg: NIST*, 2002.

[17] Hmeidi, I., Kanaan, G. and M. Evens (1997) Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. *Journal of the American Society for Information Science*, 48/10, pp. 867-881.

[18] Hull, D. Using Statistical Testing in the Evaluation of Retrieval Performance. *In Proceedings of the 16th ACM/SIGIR Conference*, pages 329-338, 1993.

[19] Khoja, S. and Garside, R. Stemming Arabic text. *Computing Department, Lancaster University*, Lancaster, 1999.

[20] Kadri, Y., and Nie, J. Y. (2006), Effective stemming for Arabic information retrieval". The challenge of Arabic for NLP/MT Conference, The British Computer Society. London, UK.

[21] Larkey, Leah S., Ballesteros, Lisa, and Connell, Margaret. (2002) Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. *In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002)*, Tampere, Finland, August 11-15, 2002, pp. 275-282.

[22] Larkey, S. L., Ballesteros, L., and Connell, E. M. (2005), Light stemming for Arabic information retrieval. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*.

[23] LDC, Linguistic Data Consortium. LDC2001T55, 2001. Available from: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T55 (accessed 1 August 2011)

[24] LDC, Linguistic Data Consortium. Buckwalter Morphological Analyzer Version 1.0, LDC2002L49, 2002. Available from: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49, (accessed 1 August 2011)

[25] Mansour N., Ramzi A. Haraty, Walid Daher, and Manal Houri. 2008. An auto-indexing method for Arabic text. *Inf. Process. Manage.* 44, 4 (July 2008), 1538-1545.
[26] Nwesri A., S.M.M Tahaghoghi, Falk Scholer, Stemming Arabic Conjunctions and Prepositions, In Mariano Consens and Gonzalo Navarro (eds.), *Lecture Notes in Computer Science - Proceedings of the Twelfth International Symposium on String Processing and Information Retrieval (SPIRE'2005)*, Buenos Aires, Argentina, 3772:206-217, November 2-4,2005.
[27] Nwesri A., S.M.M. Tahaghoghi and Falk Scholer, Arabic Text Processing for Indexing and Retrieval, *Proceedings of the International Colloquium on Arabic Language Processing*, Rabat, Moroc, 18-19 June, 2007.
[28] Taghva, K., Elkoury, R., and Coombs, J. Arabic Stemming without a root dictionary. 2005.
[29] Wonnacott, R., Wonnacott, T. *Introductory Statistics, John Wiley & Sons, Fourth Edition*,1990.
[30] Lemur Project Website: http://www.lemurproject.org/ (accessed 1 August 2011)