Semantic Query Expansion for Arabic Information Retrieval

Ashraf Y. Mahgoub
Computer Engineering Department, Cairo University, Egypt

ashraf.thunderstorme@gmail.com

Mohsen A. Rashwan
Electronics and Communications Engineering Department, Cairo University, Egypt

mrashwan@rdi-eg.com

Hazem Raafat
Computer Science Department, Kuwait University, Kuwait City, Kuwait

hazem@cs.ku.edu.kw

Mohamed A. Zahran
Computer Engineering Department, Cairo University, Egypt

moh.a.zahran@eng.cu.edu.eg

Magda B. Fayek
Computer Engineering Department, Cairo University, Egypt

magdafayek@ieee.org

Abstract

Traditional keyword based search is found to have some limitations. Such as word sense ambiguity, and the query intent ambiguity which can hurt the precision. Semantic search uses the contextual meaning of terms in addition to the semantic matching techniques in order to overcome these limitations. This paper introduces a query expansion approach using an ontology built from Wikipedia pages in addition to other thesaurus to improve search accuracy for Arabic language. Our approach outperformed the traditional keyword based approach in terms of both F-score and NDCG measures.

1 Introduction

As traditional keyword based search techniques are known to have some limitations, many researchers are concerned with overcoming these limitations by developing semantic information retrieval techniques. These techniques are concerned with the meaning the user seeks rather than the exact words of the user's query. We consider four main features that make users prefer semantic based search systems over keyword-based: Handling Generalizations, Handling Morphological Variants, Handling Concept matches, and Handling synonyms with the correct sense (Word Sense Disambiguation).

2 Semantic-based Search Features

In this section we will discuss the main features of semantic search that makes it more tempting choice over the traditional keyword based techniques.

2.1 Handling Generalization

Handling generalizations allows the system to provide the user with pages that contains material relevant to sub-concepts of the user's query. Consider the following example in Table 1 where a query contains a general term or concept "عنف"(Violence).

October 25, 2014, Doha, Qatar.

User's Query In Arabic	Equivalent Query In English
"اعمال عنف فى افريقيا"	"Violence in Africa"

Table1: Example Query 1

Semantic-based search engines should be able to recognise pages with sub-concepts like: "ابادة"(extermination),"قمع" (suppression),"تعذيب" (torture) as relevant to user's query.

2.2 Handling Morphological Variations

Handling morphological variations allows the system to provide the user with pages that contain words derived from the same root as those in user's query. Consider the following example in Table 2.

User's Query In Arabic	Equivalent Query In English
"التطور فى الشرق الاوسط"	"Development in the Middle East"

Table2: Example Query 2

Pages that contain morphological variants of the word "التطور" (Development) such as "تَطوُّر", "تَطوير", and "تَطوُّرات" should also be considered relevant to user's query.

2.3 Handling Concept Matches

The system should also be aware of concepts or named entities that may be addressed with different words. Consider the following example in Table 3.

User's Query In Arabic	Equivalent Query In English
"مصر"	"Egypt"

Table3: Example Query 3

The term "مصر" has other equivalent expressions like ["جمهورية مصر العربية", "أرض الكنانة", "أم الدنيا"]. So documents that contain any of these expressions should be considered relevant.

2.4 Handling Synonyms With Correct Sense

Although the meaning of many Arabic words depends on the word's diacritics, most Arabic text is un-vowelized. For example, Table 4 shows the word "شعب" has more than a single meaning depending on its diacritization. System should be aware which meaning to consider for expansion.

Arabic vowelized word	English equivalent	Arabic synonyms
شَعْب	People, nation	مواطنين,أمم
شُعَب	Branches	فروع

Table4: Different senses for word "شعب"

3 Related Work

Query expansion techniques have been considered by many researchers. The most successful query expansion techniques depend on automatic relevance feedback with no consideration of semantic relations.

(Jinxi Xu and Ralph, 2001) used the highest TF-IDF 50 terms extracted from the top 10 retrieved documents from AFP (i.e. the TREC2001 corpus). These 50 terms where weighted due to their TF-IDF scores and added to the original query -with addition to terms from other thesaurus-with the following formula:

$$weight(t) = oldWeight(t) + 0.4 \times \sum_{\forall\, t,D} TFIDF(t, D)$$

Where D is the top retrieved documents and t is the original term. Larkey and Connell (2001) used a similar technique, but with a different scoring method.

Wikipedia has been considered as an ontology source by many researchers. This is due to its large coverage, up-to-date, and domain independency. As in (Alkhalifa and Rodrguez, 2008), they proposed an automatic technique for extending Named Entities of Arabic WordNet using Wikipedia. They depended mainly on Wikipedia's "redirect" pages and Cross-Lingual links. Also a large scale taxonomy from Wikipedia deriving technique was proposed by (Pozetto and Strube, 2007).

(Abouenour et al., 2010) proposed a system that uses Arabic WordNet to enhance Arabic question/answering. Synonyms from WordNet are used to expand the question in order to extract the most semantically relevant passages to the question.

(Milne et al., 2007) proposed a system called "KORU" for query expansion using Wikipedia's most relevant articles to user's query. The system allows the user to refine the set of Wikipedia pages to be used for expansion. KORU used "Redirect" pages for expansion; "Hyper Links" and "Disambiguation Pages" to disambiguate unrestricted text.

Our proposed system differs from KORU in several points:

(1) Adding "Subcategories" to handle generalization.
(2) Adding Wikipedia "Gloss" – First phrase of the article – when there is no "Redirect" pages available.
(3) Allowing the user to either expand all terms in a single query, or expand each term separately producing multiple queries. The result lists of these multiple queries are then combined into a single result list.
(4) Adding terms from another two supportive thesaurus, namely "Al Raed" dictionary and our constructed "Google_WordNet" dictionary.

4 Proposed System

4.1 Arabic Resources

We depend in our query expansion mechanism on three Arabic resources: (1) Arabic Wikipedia Dump, (2) "Al Raed" Dictionary. (2) "Google_WordNet" Dictionary.

4.1.1 Arabic Wikipedia

Our system depends mainly on Arabic Wikipedia as the main semantic information source. According to Wikipedia, the Arabic Wikipedia is currently the 23rd largest edition of Wikipedia by article count, and is the first Semitic language to exceed 100,000 articles.

We were able to extract 397,552 Arabic Semantic set, with 690,236 collocations. The term "Semantic Set" stands for a set of expressions that refer to the same Meaning or Entity. For example, the following set of concepts forms a semantic set for "بريطانيا" (Britain): ['المملكة المتحدة لبريطانيا', 'بريطانيا العظمى', 'أنكلترة', 'المملكة المتحدة لبريطانيا العظمى وآيرلندا', 'العظمي'].

To extract the semantic sets, we depend on the "redirect" pages in addition to the article gloss that may contain a semantic match. This match appears in the first paragraph of the article in a bold font. The categorization system of Wikipedia is very useful in the task of expanding generic queries in a more specified form. This is done by adding "subcategories" of the original term to the expanded terms.

4.1.2 The Al Raed Monolingual Dictionary:

The "Al Raed" Dictionary is a monolingual dictionary for modern words[1]. The dictionary contains 204303 modern Arabic expressions.

4.1.3 The Google_WordNet Dictionary

We collected all the words in WordNet, and translated them to Arabic using Google Translate. For each English word, Google Translate provides different Arabic translations for the English word each corresponds to a different sense, each sense has a list of different possible English synonyms. Using this useful information we were able to extend WordNet Synset entries into a bilingual Arabic-English dictionary that maps a set of Arabic synonyms to its equivalent set of English synonyms. The basic idea is that, two sets of English synonyms (each allegedly belongs to a different sense) can be fused together into one sense if the number of overlapping words between the two sets is two or more. Fusing two English sets together will fuse also their Arabic translations into one set, thus forming a list of Arabic synonyms matched to a list of English synonyms. Table 5 shows a sample of Google Translate for the word "tough". We can fuse the first and the fourth sense together because they have two words in common namely "strong" and "robust". The same applies to the second and the third senses with "strict" and "tough" in common.

[1] Available at http://www.almaany.com/appendix.php?language=arabic&category=الرائد&lang_name=عربي

Thus forming two new mappings as shown in Table 6.

متين	solid, **strong**, **robust**, firm, durable
صارم	strict, rigorous, tough, rigid, firm, stringent
قاسي	tough, harsh, rough, severe, strict, stern
قوي	**strong**, powerful, sturdy, **robust**, vigorous

Table 5: A sample of Google Translate result for the word "tough"

قوي ,متين	solid, strong, robust, firm, durable, powerful, sturdy, vigorous
قاسي ,صارم	strict, rigorous, tough, rigid, firm, stringent, harsh, rough, severe, stern

Table 6: Mapping between a set of Arabic synonyms to a set of English synonyms.

Finally, we use words of the same Arabic set as an expansion to each other in queries.

4.2 Indexing and Retrieval

Our system depends on "LUCENE", which is free open source information retrieval library released under the Apache Software License. LUCENE was originally written in Java, but it has ported to other programming languages as well. We use the ".Net" version of LUCENE.

LUCENE depends on the Vector Space Model (VSM) of information retrieval, and the Boolean model to determine how relevant a given Document is to a User's query. LUCENE has very useful set of features, as the "OR" and "AND" operators that we depend on for our expanded queries. Documents are analyzed before adding to the index on two steps: diacritics and stop-words Removal, and text Normalization. A list of 75 words (Contains: Pronouns, Prepositions…etc.) has been used as stop-words.

4.2.1 Normalization

Three normalization rules were used:

- Replace "إ" with "ي".
- Replace "ا", "آ", "أ" with "ا"
- Replace "ه" with "ة"

4.2.2 Stemming

We implemented Light-10 stemmer developed by Larkey (2007), as it showed superior performance over other stemming approaches.

Instead of stemming the whole corpus before indexing, we grouped set of words with the same stem and found in the same document into a dictionary, and then use this dictionary in expansion. This reduces the probability of matching between two words sharing the same stem but with different senses, as they must be found in the same document in corpus to be used in expansion.

Consider the following example in table 7:

Arabic Word	Stem	English Equivalent
الطاعة	طَاعَ	Obedience
الطاعون	طَاع	Plague

Table 7: Example of two words sharing the same stem but have different senses.

We see that both words share the same stem "طاع", yet we don't expand the word "طاعة" with the word "الطاعون" as there is no document in the corpus that contains both words.

4.3 Query Expansion

To expand a query, we first locate named entities or concepts that appear in the query in Wikipedia. If a named entity or a concept has been located, we add title of "redirect" pages that leads to the same concept in addition to its subcategories from Wikipedia's categorization system. If not, we depend on the other two dictionaries –Al Raed and Google_WordNet- for expansion.

We investigated two methodologies for query expansion; the first is the most common query expansion methodology which is to produce a single expanded query that contains all expanded terms. The second methodology we introduced is to expand each term one at a time producing multiple queries, and then combine the results of these queries into a single result list. The second methodology was found less sensitive to noise

because for each expanded query, there is only one source of noise which is the term being expanded, while other terms are left without expansion. It also allows the system to boost documents from one expanded query over other documents according to the relevancy score of the expanded term.

The following example explains this intuition:
For the query "أحكام الأضاحي"
Single Expanded Query:

(أحكام OR احكام OR حكم) (الأضاحي OR الاضاحي OR إضحية OR أضاحي OR ليلة إضحية مضيئة OR ضحو OR شاة يضحى بها)

Multiple Expanded Queries:

1-(أحكام OR احكام OR حكم) الأضاحي
2- أحكام (الأضاحى OR الاضاحى OR إضحية OR أضاحي OR ليلة إضحية مضيئة OR ضحو OR شاة يضحى بها)

We see that the term "أحكام" gets fewer expansions than the term "الأضاحي"; this is because the term "الأضاحي" is less frequent in the corpus thus it needs more expansions. We then combine the results of the two queries by the following algorithm:

1- Foreach expanded query Q_i
    a. Foreach retrieved document DQ_i for Q_i
    b. If the final list contains DQ_i increment the score of DQ_i by $RF[tQ_i] \times Score(Q_i, DQ_i)$
    c. Else add DQ_i to final list

Where RF is a list of relevancy factors calculated for each term in the original query. This factor depends on the term frequency in corpus. RF is calculated according to the following formula:

$$RF[t] = \frac{1}{\log(frequency[t] + 0.5 \times \log(frequency[stemmed_t])}$$

Where t is the term we need to calculate its relevancy score, $frequency[t]$ is the numbers of times the term t appeared in the corpus, and $frequency[stemmed\_t]$ is the number of times words that share the same stem of the term appeared in the corpus. Then we sort the final list in ascending order according to their scores.

Note that the multiple expanded queries methodology consumes more time over the single expanded query. This is because each expanded query is sent to LUCENE separately. Then we combine the returned documents lists of the queries into a final documents list.

We also limit the maximum number of added terms for each term in order to reduce the noise effect of query expansion step; this maximum number also depends on the term's relevancy factor. We set the maximum number of added terms to a single query to 50. Each term gets expanded with number of terms proportional to its relevancy score. This also increases the recall as less frequent terms gets expanded more times than most frequent terms, allowing LUCENE to find more relevant pages for infrequent terms.

5 Experiments

For testing our system, we used a data set constructed from "Zad Al Ma'ad" book written by the Islamic scholar "Ibn Al-Qyyim". The data set contains 25 queries and 2730 documents. Titles of the book chapters are used as "Queries" and sections of each chapter are used as set of relevant documents for that query. Each query is tested against the whole sections.

The following tables show the values of precision, recall, f-score, and NDCG (Normalize Discounted Cumulative Gain) of three runs.
R1: No expansion is used (base line).
R2: Single expanded query.
R3: Multiple expanded queries methodology.

	R1	R2	R3
Precision @1	0.68	0.6	**0.72**
Precision @5	0.504	**0.576**	0.568
Precision @10	0.38	0.436	**0.444**
Precision @20	0.268	0.3	**0.326**
Precision @30	0.2038	0.232	**0.2546**

Table 8: Levels of Precision

	R1	R2	R3
Recall @1	0.1346	0.1067	**0.1361**
Recall @5	0.3258	**0.35721**	0.3465
Recall @10	0.3908	0.4292	**0.4390**
Recall @20	0.4804	**0.5487**	0.5393
Recall @30	0.5089	0.5806	**0.5944**

Table 9: Levels of Recall

	R1	R2	R3
F-score @1	0.1919	0.1535	**0.1948**
F-score @5	0.3249	**0.3635**	0.3528
F-score @10	0.3067	0.3466	**0.3516**
F-score @20	0.2701	0.3122	**0.3243**
F-score @30	0.2334	0.2697	**0.2868**

Table 10: Levels of F-Score

	R1	R2	R3
NDCG @1	0.68	0.6	**0.72**
NDCG @5	0.8053	**0.8496**	0.8349
NDCG @10	0.7659	0.8304	**0.8316**
NDCG @20	0.7392	0.7993	**0.8186**
NDCG @30	0.7323	0.7944	**0.8001**

Table 11: Levels of NDCG

6 Conclusion

In this paper we introduced a new technique for semantic query expansion using a domain independent semantic ontology constructed from Arabic Wikipedia. We focused on four features for semantic search: (1) Handling Generalizations. (2) Handling Morphological Variants. (3) Handling Concept Matches. (4) Handling Synonyms with correct senses. We compared both single expanded query and multiple expanded queries approaches against the traditional keyword based search. Both techniques showed better results than the base line. While the Multiple Expanded Queries approach performed better than Single Expanded Query in most levels.

7 ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their constructive comments and suggestions. This work was supported by RDI© (http://www.rdi-eg.com/)

8 References

David Milne Ian H. Witten David M. Nichols. 2007. A knowledge-based search engine powered by Wikipedia. Conference on Information and Knowledge Management (CIKM).

Jinxi Xu, Alexander Fraser and Ralph Weischedel. 2001. Cross-lingual Retrieval at BBN. TREC10 Proceedings.

Lahsen Abouenour, Karim Bouzouba, and Paolo Rosso. 2010. An evaluated semantic query expansion and structure-based approach for enhancing Arabic question/answering. International Journal on Information and Communication Technologies.

Leah S. Larkey and Margaret E. Connell. 2001. Arabic Information Retrieval at UMass. TREC10 Proceedings.

Leah S. Larkey and Lisa Ballesteros and Margaret E. Connell. 2007. Arabic Computational Morphology Text, Speech and Language Technology.

Musa Alkhalifa and Horacio Rodrguez. 2008. Automatically Extending Named Entities coverage of Arabic WordNet using Wikipedia. International Journal on Information and Communication Technologies.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from Wikipedia. AAAI'07 Proceedings of the 22nd national conference on Artificial intelligence.