Query Term Expansion by Automatic Learning of Morphological Equivalence Patterns from Wikipedia

Kareem Darwish Ahmed M. Ali Ahmed Abdelali

Qatar Computing Research Institute
Doha, Qatar

{kdarwish, amali, aabdelali}@qf.org.qa

ABSTRACT

Retrieval in many languages would benefit from language-specific processing, such as stemming or morphological analysis. However, many languages lack such processing tools, or they may be inadequate for retrieval due to language evolution. In this paper, we explore the use of Wikipedia redirects to automatically learn morphological equivalence patterns. Character-level alignment of automatically found morphological variants from Wikipedia redirects is used to generate character-level transformations. Then, given a query word, character-level transformations are used to produce morphological equivalents. The proposed method is language independent and can be applied to new languages without need for linguistic knowledge. Though, the performance of this approach may in the aggregate lag behind state-of-the-art stemming (or morphological analysis) for languages with good existing processors, the approach is generally safer than stemming in the sense that if it degrades queries, the degradation is generally marginal. Stemming on the other hand can significantly degrade queries. We show its success for Arabic, English, Hungarian, and Portuguese.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: *Query formulation;* H.2.4 [**Systems**]: Query Processing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms

Measurement, Experimentation, Languages.

Keywords

Query Expansion; Morphological Analysis; Inflection; Information Retrieval.

1. INTRODUCTION

Many languages exhibit rich morphological phenomena that complicate retrieval. For example, Arabic has a derivational morphology that allows for the attachment of prefixes, suffixes, and infixes. Morphology is further complicated when new words are borrowed from other languages, where morphological rules are applied to borrowed words. Consider the English word "topic", which is often transliterated as "توبك" (twbk). Since it is borrowed, it is unclear whether it is feminine or masculine. Thus, it is often made into a plural using the feminine plural by adding the suffix "ات" (At) to make it "توبيكات" (twbykAt) or into a broken plural "توابك" (twAbk). Many languages lack good morphological analyzers that can handle morphology properly, particularly for retrieval. Even for languages with good analyzers, the analyzers may not be able to handle borrowed words properly. Also, incorrect morphological analyses or improper stemming may adversely affect retrieval effectiveness.

1.1 Problem Statement

A user issuing a query typically provides just one morphological form of any given word. In this paper, we address the broad problem of generating morphological variations of query terms to improve retrieval effectiveness. Specifically, we want to generate such variants in a language-independent way to avail the need for linguistic resources that may be lacking for many languages. We seek to show that using language-independent methods can perform at par with linguistically motivated methods. The method can also handle borrowed words as in the example above.

1.2 Proposed Solution

In this paper, we explore the use of Wikipedia page redirects to learn morphological character-level transformations to generate morphological equivalents of query terms. Such terms are used for query expansion. Essentially, a redirect page is an empty Wikipedia page that automatically redirects a user's request for a particular title (ex. accessible computing) to another content page with a synonymous title (ex. computer accessibility). The title of a redirect page and the title of the corresponding content page are typically parallel. Redirects handle cases such as:

Alternative names:
Computer_games ⇒ Video_game

Alternative spellings:
Chang San-feng ⇒ Zhang Sanfeng

Common misspellings:
Condaleeza Rice ⇒ Condoleezza Rice

Closely related words:
Communists ⇒ Communism

Abbreviations:
CDMA ⇒ Code Division Multiple Access

Initially, we extract potential word variations from the parallel redirect-content page titles using simple string similarity. Then, given a large set of potential word variations, we align the pairs at character level to learn possible character-level transformations. Then, given a new query word, the induced transformations are applied to it, while restricting the equivalents to words that appear in a large word list. We use such equivalents at query time to improve retrieval effectiveness. We apply the proposed technique on Arabic, English, Hungarian, and Portuguese retrieval. The choice of languages is motivated by several factors, namely:

- Morphology: where Arabic and Hungarian have rich morphologies and English and Portuguese have simpler morphologies.
- Collection size: where the English collection is very large, the Arabic and Portuguese collections are medium sized, and the Hungarian collection is relatively small.
- Number of available Wikipedia redirects: where English has millions, Arabic and Portuguese have hundreds of thousands, and Hungarian has tens of thousands.

1.3 Contributions

The main contribution of this paper is the automatic learning of language-independent morphological character-level variations from Wikipedia redirects. We use the character-level variations to generate morphological variations that can be used for expanding query words (section 3). We show that the induced character-level variations statistically significantly improve retrieval effectiveness. We explore the space of possibilities by experimenting on languages with: simple or rich morphologies (derivational and inflectional), collections of varying sizes, and varying number of available Wikipedia redirects. All retrieval experiments are performed on standard datasets (section 4).

2. BACKGROUND

Concerning the automatic induction of morphological variations, Hammarström [11] surveyed fairly comprehensively many unsupervised morphology learning approaches. Brent et al. [2] proposed the use of Minimum Description Length (MDL) to automatically discover suffixes. MDL based approach was improved by: Goldsmith [9] who applied the Expectation-Maximization (EM) algorithm to improve the precision of pairing stems prior to suffix induction; and Schone and Jurafsky [24] who applied latent semantic analysis to determine if two words are semantically related prior to suffix induction. Jacquemin [12] used word grams that look similar, i.e. share common stems, to learn suffixes. Baroni [1] extended his work by incorporating semantic similarity features, via mutual information, and orthographic features, via edit distance. Chen and Gey [3] utilized a bilingual dictionary to find Arabic words with a common stem that map to the same English stem. Also in the cross-language spirit, Snyder and Barzilay [25] used cross-language mappings to learn morpheme patterns and consequently automatically segment words. They successfully applied their method to Arabic, Hebrew, and Aramaic. Creutz and Lagus [4] proposed a probabilistic model for automatic word segment discovery. Most these approaches can discover suffixes and prefixes without human intervention. However, they may not be able to handle infixation and spelling variations. Karagol-Ayan et al. [14] used approximate string matching to automatically map morphological variant in noisy dictionary data. They used the mappings to learn affixation, including infixation, from noisy data. In this paper, we propose a new technique for finding morphological variations based on learning character-level mappings.

Arabic has a rich derivational morphology where words typically are derived from a set of a few thousand roots. A root is fit into a template that may include prefixes, suffixes, and infixes to generate stems. Arabic orthography is complicated by optional diacritics and pronouns, prepositions, determiners, and coordinating conjunctions that are attached to stems. Most recent studies on Arabic retrieval were based on a single, large collection (from TREC-2001/2002) [8, 20]. Removing diacritics and conflating some Arabic characters improved retrieval effectiveness [27]. Using linguistically informed in-context stemming [6] and light stemming [16] seemed to produce the best results. Diab [8] and Lee et al. [17] proposed systems for performing in-context Arabic stemming. We use the system proposed by Diab [8] for comparison in this paper.

English and Portuguese have relatively non-complex inflectional morphology. The literature on the effect of stemming on English retrieval is too large to cite here. The effect of stemming on English retrieval is collection and genre dependent, without any general rule on the usefulness of stemming for retrieval. For Portuguese, Orengo et al. [21] suggested that light stemming, where the plural word forms are stemmed, seemed to work best, but improvements due to stemming were relatively small (around 8% relative improvement over using words). Savoy [23] noted similar results with slightly less than 8% improvements in Portuguese retrieval effectiveness when using a Porter-like stemmer.

Hungarian has a rich inflectional morphology, where words may have prefixes, suffixes, and circumfixes, which are combinations of prefixes and suffixes. Halácsy and Trón [10] and Savoy [23] suggested that Hungarian retrieval benefits significantly from stemming, with more than 40% increase in retrieval effectiveness, as measured by mean average precision. However, all the reported Hungarian experiments were conducted on a very small collection of less than 50,000 documents [19].

In small collections, improvements in recall typically yield to great improvements in retrieval effectiveness. However, for larger collections, improvements in recall may adversely affect overall retrieval effectiveness. Generally, the larger a retrieval collection gets, the lesser language processing is required for effective retrieval, and vice versa. In this work, we experiment on collections of varying sizes to ascertain the effectiveness of the proposed technique under different conditions.

3. FINDING EQUIVALENTS

In our experiments, we extracted synonymous pairs from Wikipedia redirects. As noted earlier, a redirect is an empty page that automatically redirects a user's request for a particular title to another content page with a synonymous title. We obtained roughly 100k, 4.5M, 56k, and 464k redirect-content title pairs for Arabic, English, Hungarian, and Portuguese respectively. Next, we attempted to find word pairs that were potential morphological variations in parallel title pairs. To do so, given a pair of synonymous titles, we assumed that a word in the first title and another in the second title are variants if they matched the following criteria:

Edit distance (ED) must be < 3. The choice of 3 was motivated by the fact that Arabic prefixes and suffixes are typically 1, 2, or 3 letters long.

- Longest common substring (LCS) > 2.
- Letters in common (LIC) (in order) > 3. LIC and LCS were used in combination to allow for infixes.
- LIC > ED

For illustration, given the pair "Jon" and "John":

ED = 1 (insertion of "h")
LCS = 2 ("Jo")
LIC = 3 ("J", "o", and "n")

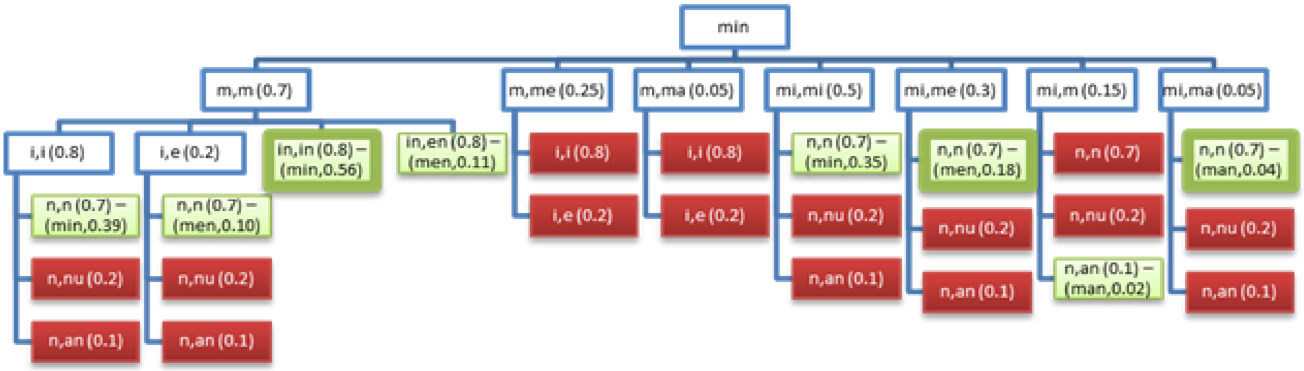

Figure 1 Decoding example. Shaded light green are final nodes producing a proper word; thick green border indicates highest probability path; red node indicates pruning for not producing valid alternative.

We experimented with other values for ED, LIC, and LCS on Arabic and English development sets, and the picked numbers seemed to subjectively produce the best candidate pairs. Given an example of parallel titles {accessible computing, computer accessibility}, "computer" and "computing" were assumed to be variants ("accessible" and "accessibility" did not pass the ED threshold).

Doing so, we obtained approximately 158k, 840k, 17k, and 184k for Arabic, English, Hungarian, and Portuguese word pairs respectively. They included primarily morphological variants and to some extent spelling variations. We aligned the word pairs at character level using Giza++ and the phrase extractor and scorer from the Moses machine translation package [15]. To apply a machine translation analogy, we treated words as sentences and the letters from which were constructed as tokens. The alignment produced letter sequence mappings. Source character sequence lengths were restricted to 3 letters.

We used the letter sequence mappings to produce morphological variations of words. We treated the problem of generating variants like a mining problem akin to that in [5]. Briefly, the miner used character segment mappings to generate all possible "transliterations" while constraining generation to the existing words in a list of unique words. For Arabic and English, we used the unique words in Wikipedia, which were 806k and 646k tokens respectively. For Hungarian and Portuguese, we used the unique tokens in the retrieval collections, which were 663k and 658k respectively.

Basically, given a query word, all possible segmentations, where each segment has a maximum length of 3 characters, were produced along with their associated mappings. Given all mapping combinations, combinations producing valid target words were retained and sorted according to the product of their mapping probabilities. To illustrate how this works, consider the following example: Given a query word "min", target words in the word list {moon, men, man, min} and the possible mappings for the segments and their probabilities:

m = {(m, 0.7), (me, 0.25), (ma, 0.05)}
mi = {(mi, 0.5), (me, 0.3), (m, 0.15), (ma, 0.05)}
i = {(i, 0.8), (e, 0.2) }
n = {n, 0.7), (nu, 0.2), (an, 0.1)}
in = {(in, 0.8), (en, 0.2)}

Figure 1 illustrates the decoding process to produce valid output words, where the probability of the output words is just the product of mapping probabilities producing the output. Consequently, the algorithm would produce the following candidates with the corresponding channel probabilities: {(min:0.56), (men:0.18), (man:0.04)}. The actual implementation of the decoder incorporates other optimizations and is described in greater detail in [13]. For example, the target word list was stored in a suffix tree to determine if a path was valid; if multiple paths led to the same output, the most probable path was pursued and all others were pruned; and paths with longer n-grams were pursued first as they were likely to have higher probabilities.

4. EVALUATING EQUIVALENTS

4.1 Experimental Setup

We used an extrinsic IR evaluation to determine the goodness of the generated equivalents. We tested on four languages as follows:

Language	Morphology	Collection Size	No. of Redirects
Arabic	Rich Derivational	Medium 383k docs	Medium 100k
English	Simple Inflectional	Very Large 50M docs	Large 4.5M
Hungarian	Rich Inflectional	Small 50k docs	Small 56k
Portuguese	Simple Inflectional	Medium 211k docs	Medium 464k

All experiments were performed using the Indri retrieval toolkit with default settings. Indri uses a retrieval model that combines inference networks and language modeling and implements state-of-the-art query operators [18]. We used a paired 2-tailed t-test with p-value less than 0.05 to determine if a set of retrieval results was better than another.

For all languages, given each query word, it is replaced with all the generated equivalents using weighted synonym operator [7][26], where the weights correspond to the product of the mapping probabilities for each equivalent word. For example, given the Arabic word "AlkrdstAny" (the Kurdish), it was replaced with:

#wsyn(0.016 krdstAnyh 0.022 krdstAn 0.043 AlkrdstAnyh 0.224 krdstAny 0.587 AlkrdstAny).

In this Arabic example, all the generated equivalents were morphological variations. Given the English word "invented", it was replaced with:

#wsyn(0.002 inventions 0.053 inventing 0.126 invention 0.579 invented).

4.1.1 Arabic

For Arabic, we used the Text REtrieval Conference (TREC) 2002 cross language track collection, which contains 383,872 Arabic newswire articles and 50 topics with relevance judgments [20]. This is presently the best available large Arabic information retrieval test collection. Since all relevance judgments were binary (relevant = 1; not-relevant = 0), we elected to use Mean Average Precision (MAP) as the measure of goodness for this retrieval task. Going down from the top a retrieved ranked list, Average Precision (AP) is the average of precision values computed at every relevant document found. MAP is just the mean of the AP's for all queries.

We used two baselines to compare to the generated equivalents, namely: using raw words, and using state-of-the-art context sensitive stemming [8]. The stemmer was trained using 400,000 manually stemmed words from the Arabic Treebank. We performed simple Arabic letter conflation, where we conflated: variants of the letter "alef", "ta marbouta" and "ha", "alef maqsoura" and "ya", and the different variants of "hamza".

4.1.2 English

For English, we used the ClueWeb09 category B collection that was used for the TREC Web and Relevance Feedback tracks in 2009 and 2010. The collection contains 50 million English pages. We used the 50 topics from the TREC 2010 Web track. The relevance judgments were made on a 5 point scale, namely 2, 1, 0, -1, and -2 corresponding to perfect, excellent, good, poor, and not relevant respectively. Therefore, we elected to use normalized Discounted Cumulative Gain (nDCG), which attempts to measure how much information a user would gain if the user starts to read from the top of the ranked list, normalized by the maximum attainable gain. We used nDCG @ 1, 3, and 10, which in web search respectively represent: the first result, which is the most likely result a user may click on; the results that typically appear on the first search screen without scrolling; and the results that appear on the first page, which users rarely go beyond.

We did not expand stopwords, where the stopword list was obtained from NLTK[1]. We used two baselines, namely: using raw words, and using stems that were generated using the Porter stemmer [22].

4.1.3 Hungarian and Portuguese

For Hungarian, we used the Cross Language Evaluation Forum (CLEF) Hungarian dataset. The dataset is composed of 49,530 documents and 50 queries (CLEF-2007 queries: 401-450) with associated binary relevance judgments.

For the Portuguese, we used the CLEF Portuguese dataset, which is composed of 210,734 documents and 50 queries (CLEF-2006 queries: 301-350) with associated binary relevance judgments. Since relevance judgments were binary for both languages, we used MAP as the measure of retrieval effectiveness. For both languages, we used two baselines, namely: using raw words, and using stems that were generated Snowball stemmer2.

4.2 Experimental Setup

4.2.1 Arabic

Table 1 reports on Arabic results. As can be seen, using stem and the proposed expansion method improved retrieval effectiveness and the improvements were statistical significant. Though using stems yielded higher MAP than the proposed expansion method, the difference was ***not*** statistically significant with t-test p-value = 0.25.

Table 1. Arabic results for MAP

	Words	Stems	Expansion
MAP	0.199	0.237	0.211

Table 2. Increase/decline in MAP (basis points) using stems and proposed expansion over using words

	No. improved	Avg. improvement	No. hurt	Avg. decline
Stems	33	0.098	14	0.094
Expansion	30	0.026	12	0.014

Table 2 reports how often stemming and expansion improved or hurt retrieval effectiveness over using words and by how much on average. Both methods improved and hurt similar numbers of queries. However, stemming either improved by a significant amount or hurt by an equally significant amount. The proposed expansion conservatively improved effectiveness, and improvements outweighed declines in effectiveness. This is generally desirable because users' reaction to adversely affected results is far greater than to positively improved results.

4.2.2 English

Table 3 reports on English results. Using stems generally degraded retrieval effectiveness. This degradation of retrieval effectiveness was expected for two reasons:

a. The English collection is relatively large, and retrieval effectiveness on larger collections is typically adversely affected by processing that is intended to increase recall.

b. Stemming conflates all inflected forms of a word to a single stem form, giving them equal weight, without regard to the likelihood of their mappings to each other. For example, it is sensible to conflict "booking" and "booked" (as in booking a ticket) together, while it should not be conflated with "books".

On the other hand, the proposed expansion averts some of the stemming problems by assigning confidence weights to the likelihood of mapping of one inflected form to another. Thus, it consistently improved retrieval effectiveness with statistically significant improvement over using words for nDCG@10.

[1] http://www.nltk.org

[2] http://snowball.tartarus.org

Table 3. English results for nDCG@{1,3,10}

nDCG@	Words	Stems	Expansion
1	0.116	0.088	0.122
3	0.105	0.092	0.106
10	0.107	0.097	0.117

Table 4. Absolute increase and decline in nDCG over using words. Unshaded for proposed expansion; shaded for stemming

nDCG@	No. Improved		Average improvement		No. Hurt		Average decline	
1	2	0	14.2	-	0	2	-	66.7
3	2	0	2.4	-	0	2	-	31.2
10	14	0	3.7	-	3	5	2.6	9.2

Table 4 reports on how often the proposed expansion (unshaded) and stemming (shaded) improved or hurt retrieval effectiveness over using just words along with the average improvement and decline in nDCG basis points. As the results show, using stemming rarely improved retrieval effectiveness. When stemming hurt retrieval, declines in nDCG were typically large. For nDCG@{1,3,10}, though the proposed technique benefited a limited number of queries, it rarely hurt retrieval effectiveness. This is desirable for the reason stated earlier.

4.2.3 Hungarian

Table 5 reports on the results for Hungarian. The proposed expansion statistically significantly improved retrieval effectiveness over the use of words. Further, using stemming statistically significantly improved retrieval effectiveness over the use of words and the proposed expansion. The considerable success of stemming over the proposed method could be attributed to the following possible reasons:

a. The number of training example to the proposed method was rather small with less than 17,000 training examples, leading to unobserved phenomena in the training examples.

b. The retrieval collection is rather small, making it benefit more from stemming, which generally improves recall.

Table 5. Hungarian results

	Words	Stems	Expansion
MAP	0.169	0.259	0.208

Table 6. Increase/decline in MAP (basis points) using stems and proposed expansion over using words

	No. improved	Avg. improvement	No. hurt	Avg. decline
Stems	39	0.120	10	0.049
Expansion	22	0.143	9	0.027

Table 6 compares how often stemming and the proposed expansion improved or hurt retrieval effectiveness over using words and by how much on average. It is clear that stemming fared much better than the proposed expansion technique for the reasons above.

4.2.4 Portuguese

Table 7 reports on results for Portuguese. Stemming performed slightly better than the proposed expansion, which in turn performed slightly better than using words, but none of the differences were statistically significant. Table 8, which compares how often stemming and the proposed expansion improved or hurt retrieval effectiveness, tells a similar story. This could be attributed to relatively simple Portuguese morphology, where gains due to stemming or expansion were relatively small. Further, the collection was small, compared to the English collection, which would favor stemming.

Table 7. Portuguese results

	Words	Stems	Expansion
MAP	0.1804	0.2087	0.1919

Table 8. Increase/decline in MAP (basis points) using stems and proposed expansion over using words

	No. improved	Avg. improvement	No. hurt	Avg. Decline
Stems	21	0.098	14	0.045
Expansion	20	0.050	12	0.042

4.3 Observations

There are a few observations worth noting:

1. For relatively small collections, as in the case of Hungarian, stemming outperforms the proposed expansion due to the gain achieved by stemming. The opposite is true for very large collections, as in the case of English.

2. The proposed technique is dependent on the amount of training examples that we were able to extract from Wikipedia. With more training examples, as in Arabic, the proposed technique was competitive with state-of-the-art context sensitive stemming. With fewer training example, as in the case of Hungarian, some statistically significant gains are possible, but stemming may do better.

3. As with stemming, the proposed technique shows more gains for morphologically rich languages, with either inflectional, like Hungarian or derivational morphology, like Arabic, compared to languages with simpler morphologies.

4. In the presence of lots of training examples and a large retrieval collection, as in the case of English, the proposed technique has the potential of delivering statistically significant gains even for languages with relatively simple morphologies.

5. Conclusion

In this paper, we presented a language independent method for rapidly learning morphological variations from Wikipedia redirects. We showed how the method is able to find morphological variations for morphologically rich language, such as Arabic and Hungarian, and morphologically less rich languages, such English and Portuguese.

The proposed method performed at par with state-of-the-art context sensitive Arabic stemming. Further, it led to statistically significant improvement in English retrieval, outpacing words and stems. Generally, the method: benefited from larger sets of Wikipedia redirects; had greater impact on morphologically rich languages; and improved retrieval for very large collections, for which stemming may adversely affect retrieval effectiveness.

For future work, it makes sense to enhance the proposed method by incorporating contextual and language modeling features to further improve morphological and spelling variants generation. This can be helpful in pruning noisy generated candidates. We would also like to try our method for other problems such detecting spelling variations of proper names.

6. References

[1] M. Baroni, J. Matiasek, H. Trost (2002). Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. ACL-2002 Workshop on Morphological & Phonological Learning, pp. 48-57.

[2] M. Brent, S. Murthy, A. Lundberg (1995). Discovering Morphemic Suffixes: A Case Study in Minimum Description Length Induction. 15th Annual Conference on the Cognitive Science Society, pp. 28-36.

[3] A. Chen, F. Gey (2002). Building an Arabic Stemmer for Information Retrieval. TREC-2002.

[4] M. Creutz, K. Lagus (2007). Unsupervised models for morpheme segmentation and morphology learning. Speech and Language Processing, Vol. 4, No 1:3, 2007.

[5] K. Darwish (2010). Transliteration Mining with Phonetic Conflation and Iterative Training. ACL NEWS Workshop, 2010.

[6] K. Darwish, H. Hassan, O. Emam (2005). Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. ACL Workshop on Computational Approaches to Semitic Languages, pp. 25–30, 2005.

[7] K. Darwish, D. Oard (2003). Probabilistic Structured Query Methods. In ACM SIGIR: 338-344, 2003.

[8] M. Diab (2009). Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. 2nd Int. Conf. on Arabic Language Resources and Tools, 2009.

[9] J. Goldsmith (2001). Unsupervised Learning of the Morphology of a Natural Language. Journal of Computational Linguistics, Vol. 27:153-198, 2001.

[10] P. Halácsy, V. Trón (2006). Benefits of Resource-Based Stemming in Hungarian Information Retrieval. CLEF 2006, LNCS 4730, pp. 99–106, 2007.

[11] H. Hammarström (2009). Unsupervised Learning of Morphology and the Languages of the World. Ph.D. Thesis, Dept. of CSE, Chalmers Univ. of Tech. and Univ. of Gothenburg.

[12] C. Jacquemin (1997). Guessing morphology from terms and corpora. ACM SIGIR-1997, v.31 n.SI, p.156-165.

[13] A. El-Kahky, K. Darwish, A. Saad Aldein, M. Abd El-Wahab, A. Hefny, W. Ammar (2009). Improved Transliteration Mining Using Graph Reinforcement. EMNLP-2011, 2011.

[14] B. Karagol-Ayan, D. Doermann, A. Weinberg (2006). Morphology Induction from Limited Noisy Data Using Approximate String Matching. 8th ACL SIG on Computational Phonology at HLT-NAACL 2006, pp. 60–68.

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst (2007). Moses: Open Source Toolkit for Statistical Machine Translation, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007.

[16] L. Larkey, L. Ballesteros, and M. Connell (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. SIGIR 2002. pp. 275-282.

[17] Y. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan (2003). Language Model Based Arabic Word Segmentation. In the Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, July 2003, Sapporo, Japan. p. 399 - 406.

[18] D. Metzler, W. B. Croft (2004). Combining the Language Model and Inference Network Approaches to Retrieval. Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval, 40(5), 735-750, 2004.

[19] G. Nunzio, N. Ferro, T. Mandl, C. Peters (2007). CLEF 2006: Ad Hoc Track Overview. CLEF 2006, LNCS 4730, pp. 21–34, 2007.

[20] D. Oard, F. Gey (2002). The TREC 2002 Arabic/English CLIR Track. In TREC 2002.Gaithersburg, MD.

[21] V. M. Orengo, L. S. Buriol and A. R. Coelho (2007). A Study on the Use of Stemming for Monolingual Ad-Hoc Portuguese Information Retrieval. CLEF 2006, LNCS 4730, pp. 91–98, 2007.

[22] M. F. Porter (1980). An algorithm for suffix stripping. Program, 14(3) pp. 130−137.

[23] J. Savoy (2007) Searching strategies for the Bulgarian language. Information Retrieval 10(6), pp. 509-529.

[24] P. Schone, D. Jurafsky (2001). Knowledge-free induction of inflectional morphologies. ACL 2001.

[25] B. Snyder, R. Barzilay (2008). Unsupervised Multilingual Learning for Morphological Segmentation. ACL-08: HLT, pp. 737–745, 2008.

[26] J. Wang, D. Oard (2006). Combining Bidirectional Translation and Synonymy for Cross-language Information Retrieval. SIGIR-2006, pp. 202-209.

[27] J. Xu, A. Fraser, and R. Weischedel (2001). 2001 Cross-Lingual Retrieval at BBN. In TREC, 2001. Gaithersburg, MD. p. 68 - 75.