Arabic Retrieval Revisited: Morphological Hole Filling

Kareem Darwish, Ahmed M. Ali
Qatar Computing Research Institute
Qatar Foundation, Doha, Qatar
kdarwish@qf.org.qa, amali@qf.org.qa

Abstract

Due to Arabic's morphological complexity, Arabic retrieval benefits greatly from morphological analysis – particularly stemming. However, the best known stemming does not handle linguistic phenomena such as broken plurals and malformed stems. In this paper we propose a model of character-level morphological transformation that is trained using Wikipedia hypertext to page title links. The use of our model yields statistically significant improvements in Arabic retrieval over the use of the best statistical stemming technique. The technique can potentially be applied to other languages.

1. Introduction

Arabic exhibits rich morphological phenomena that complicate retrieval. Arabic nouns and verbs are typically derived from a set of 10,000 roots that are cast into stems using templates that may add infixes, double letters, or remove letters. Stems can accept the attachment of clitics, in the form of prefixes or suffixes, such as prepositions, determiners, pronouns, etc. Orthographic rules can cause the addition, deletion, or substitution of letters during suffix and prefix attachment. Further, stems can be inflected to obtain plural forms via the addition of suffixes or through using a different stem form altogether producing so-called broken[1] (aka irregular) plurals.

For retrieval, we would ideally like to match "related" stem forms regardless of inflected form or attached clitic. Tolerating some form of derivational morphology where nouns are transformed into adjectives via the attachment of the suffix ي (y)[2] (ex. مصر (mSr) ➔ مصري (mSry)) is desirable as they are semantically related. Matching all stems that are cast from the same root would introduce undesired ambiguity, because a single root can produce up to 1,000 stems.

Two general approaches have been shown to improve Arabic retrieval. The first approach involves stemming, which removes clitics, plural and gender markers, and suffixes such as ي (y). Statistical stemming was reported to be the most effective for Arabic retrieval (Darwish et al., 2005). Though effective, stemming has the following drawbacks:

1. Stemming does not handle infixes and hence cannot conflate singular and broken plural word forms. For example, the plural of the Arabic word for book "كتاب" (ktAb) is "كتب" (ktb).
2. Stemming of some named entities, which are important for retrieval, and their inflected forms may produce different stems as word endings may change with the attachment of suffixes. Consider the Arabic words for America أمريكا (>mrykA) and American أمريكي (>mryky), where the final letter is transformed from "A" to "y".

The second approach involves using character 3- or 4-grams (as opposed to words) (Mayfield et al., 2001; Darwish and Oard, 2002). For example, the trigrams of "WORD" are "WOR" and "ORD". This approach though it has been shown to improve retrieval effectiveness, it has the following drawbacks:

1. It cannot handle broken plurals, though it would handle words where stemming would produce different stems for different inflected forms.
2. It significantly increases index sizes. For example, using a 6 letter word would produce 4 trigram chunks, which would have 12 letters.
3. Longer words would yield more character n-gram chunks compared to shorter ones leading to skewed weights for query words.

[1] "Broken" is a direct translation of the Arabic word "takseer", which refers to this kind of plural.

[2] We use Buckwalter transliteration in the paper

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 218–222,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

To address this problem, we propose the use of a character level transformation model that can generate tokens that are morphologically related to query tokens. We train the model using morphological related stems that are extracted from hypertext/page title pairs from Wikipedia. Such pairs are good for the task at hand, because they show different ways to refer to the same concept. We show that expanding stems in a query with related stems using our model outperforms the use of state-of-the-art statistical Arabic stemming. Further, the expansion can be applied to words directly to perform at par with statistical stemming. Laterally, the model can help produce spelling variants of transliterated names.

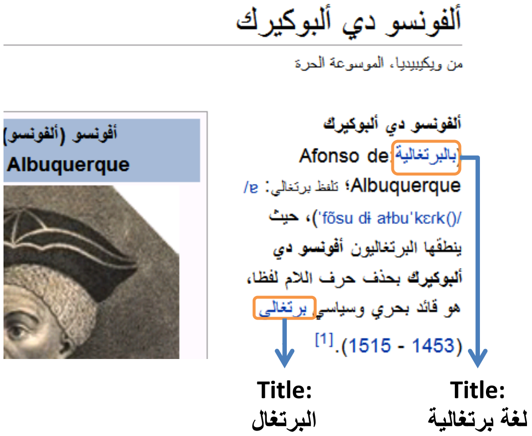
The contribution of this paper is as follows:

- We proposed an automatic method for learning character-level morphological transformations from Wikipedia hypertext/page title pairs.
- When applied to stems, we show that the method overcomes some morphological problems that are associated with stemming, statistically significantly outperforming Arabic retrieval using statistical stemming and character n-grams.
- When applied to words, we show that the method yields retrieval effectiveness at par with statistical stemming.

2. Related Work

Most studies are based on a single large collection from the TREC-2001/2002 cross-language retrieval track (Gey and Oard, 2001; Oard and Gey, 2002). The studies examined indexing using words, word clusters (Larkey et al., 2002), terms obtained through morphological analysis (e.g., stems and roots (Darwish and Oard, 2002), light stemming (Aljlayl et al., 2001; Larkey et al., 2002), and character n-grams of various lengths (Darwish and Oard, 2002; Mayfield et al., 2001). The effects of normalizing alternative characters, removal of diacritics and stop-word removal have also been explored (Xu et al., 2001). These studies suggest that light stemming, character n-grams, and statistical stemming are the better index terms. Morphological approaches assume an Arabic word is constituted from prefixes-stem-suffixes and aim to remove prefixes and suffixes. Since Arabic morphology is ambiguous, statistical stemming attempts to find the most likely segmentation of words. The first such systems were MORPHO3 (Ahmed, 2000) and Sebawai (Darwish, 2002). Later work by Lee et al. (2003) used a trigram language model with a minimal set of manually crafted rules to achieve a stemming accuracy of 97.1%. Their system was shown by Darwish et al. (2005) to lead to statistical improvements over using light stemming. Diab (2009) used an SVM classifier to ascertain the optimal segmentation for a word in context. The classifier was trained on the Arabic Penn Treebank data. She reported a stemming accuracy of 99.2%. Although consistency is more important for IR applications than linguistic correctness, perhaps improved correctness would naturally yield great consistency. In this paper, we used a reimplementation of the system proposed by Diab (2009) with the same training set as a baseline.

Concerning the automatic induction of morphologically related word-forms, Hammarström (2009) surveyed fairly comprehensively many unsupervised morphology learning approaches. Brent et al. (1995) proposed the use of Minimum Description Length (MDL) to automatically discover suffixes. MDL based approach was improved by: Goldsmith (2001) who applied the EM algorithm to improve the precision of pairing stems prior to suffix induction; and Schone and Jurafsky (2001) who applied latent semantic analysis to determine if two words are semantically related. Jacquemin (1997) used word grams that look similar, i.e. share common stems, to learn suffixes. Baroni (2002) extended his work by incorporating semantic similarity features, via mutual information, and orthographic features, via edit distance. Chen and Gey (2002) 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 (2008) 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 (2007) proposed a probabilistic model for automatic word segment discovery. Most of 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. (2006) used approximate string matching to automatically map morphologically similar words in noisy dictionary data. They used the mappings to learn affixation, including infixiation, from noisy data. In this paper, we propose a new technique for finding morphologically related word-forms based on learning character-level mappings.



ألفونسو دي ألبوكيرك

من ويكيبيديا، الموسوعة الحرة

أفونسو (ألفونسو)
Albuquerque

ألفونسو دي ألبوكيرك
Afonso de (بالبرتغالية
Albuquerque؛ تلفظ برتغالي: /ɐ
'fõsu dɨ aɫbu'kɛrk()/)، حيث
ينطقها البرتغاليون أفونسو دي
ألبوكيرك بحذف حرف اللام لفظا،
هو قائد بحري وسياسي برتغالي
(1515 - 1453).[1]

Figure 1. Example hypertexts to Wikipedia titles

3. Character-Level Model

3.1 Training Data

In our experiments, we extracted Wikipedia hypertext to page title pairs as in Figure 1. We performed all work on an Arabic Wikipedia dump from April 2010, which contained roughly 150,000 articles. In all, we extracted 11.47 million hypertext-title pairs. From them, we attempted to find word pairs that were morphologically related. From the example in Figure 1, given the hypertext بالبرتغالية (bAlbrtgAlyp – in Portuguese) and the page title that it points to لغة برتغالية (lgp brtgAlyp – Portuguese language) we needed to extract the pairs بالبرتغالية (bAlbrtgAlyp) and برتغالية (brtgAlyp).

We assumed that a word in the hypertext and another in Wikipedia title were morphologically related using the following criteria:

- The words share the first 2 letters or the last 2 letters. This was intended to increase precision.
- The edit distance between the two words 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.
- The edit distance was less than 50% of the length of the shorter of the two words. This was important to insure that short words that share common letters but are in fact different are filtered out.

The word pairs that matched these criteria were roughly 13 million word pairs[3]. All words in the word pairs were stemmed using a reimplementation of the stemmer of Diab (2009).

3.2 Alignment and Generation

<u>**Alignment:**</u> We performed two alignments. In the first, we aligned the stems of the word pairs at character level. In the second, we aligned the words of the word pairs at character level without stemming. The pairs were aligned using Giza++ and the phrase extractor and scorer from the Moses ma-chine translation package (Koehn et al., 2007). 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.

<u>**Generating related stems/words:**</u> We treated the problem of generating morphologically related stems (or words) like a transliteration mining problem akin to that in Udupa et al. (2009). Briefly, the miner used character segment mappings to generate all possible transformations while constraining generation to the existing tokens (either stems or words) in a list of unique tokens in the retrieval test collection.

Basically, given a query token, 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 tokens 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)}
n = {n, 0.7), (nu, 0.2), (an, 0.1)}
in = {(in, 0.8), (en, 0.2)}

The algorithm would produce the following candidates with the corresponding channel probabilities:

(min➔min:0.56): (m➔m: 0.7); (in➔in: 0.8)
(min➔men:0.18): (m➔m: 0.7); (in➔en: 0.2)

[3] The training data can be obtained from: https://github.com/kdarwish/WikiPairs

(min➔man:0.035): (mi➔ma: 0.05); (n➔n: 0.7) The implementation details of the decoder are described in (El-Kahki et al., 2012).

4. Testing Arabic Retrieval Effectiveness

4.1 Experimental Setup

We used extrinsic IR evaluation to determine the quality of the related stems that were generated. We performed experiments on the TREC 2001/2002 cross language track collection, which contains 383,872 Arabic newswire articles and 75 topics with their relevance judgments (Oard and Gey, 2002). This is presently the best available large Arabic information retrieval test collection. We used 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.

All experiments were performed using the Indri retrieval toolkit, which uses a retrieval model that combines inference networks and language modeling and implements advanced query operators (Metzler and Croft, 2004). 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.

We replaced each query tokens with all the related stems that were generated using a weighted synonym operator (Wang and Oard, 2006), where the weights correspond to the product of the mapping probabilities for each related word. With the weighted synonym operator, we did not need to threshold the generated related stems as ones with low probabilities were demoted. Probabilities were normalized by the score of the original query word. For example, given the stem صناع (SnAE) it was replaced with: #wsyn(1.000 SnAE 0.029 SnAEy 0.013 SnE 0.006 SnAEA 0.003 mSnwE).

We used three baselines to compare against, namely: using raw words, using statistical stemming (Diab, 2009), and character 4-grams. For all runs, we performed letter normalization, where we conflated: variants of "alef", "ta marbouta" and "ha", "alef maqsoura" and "ya", and the different forms of "hamza".

Table 1. Retrieval Results

Run	MAP	Statistically better than
Words	0.225	
Stems	0.276	words
Char 4-grams	0.244	
Expanded Words	0.264	words
Expanded Stems	0.296	words/stems/char 4-grams

4.2 Experimental Results

Table 1 reports retrieval results. Expanding stems using morphologically related stems yielded statistically significant improvements over using words, stems, and character 4-grams. Expanding words yielded results that were statistically significantly better than using words, and statistically indistinguishable from using 4-grams and stems. As the results show, the proposed technique improves upon statistical stemming by overcoming the shortfalls of stemming. Another phenomenon that was addressed implicitly by the proposed technique had to do with detecting variant spellings of transliterated names. This draws from the fact that differences in spelling variations and the construction of broken plurals are typically due to the insertion or deletion of long vowels. For example, given the name "نتنياهو" (ntnyAhw– Netanyahu), the model proposed: ntynyAhw, ntAnyAhw, and ntAnyhw.

5. Conclusion

In this paper, we presented a method for generating morphologically related tokens from Wikipedia hypertext to page title pairs. We showed that the method overcomes some of the problems of statistical stemming to yield statistically significant improvements in Arabic retrieval over using statistical stemming. The technique can also be applied on words to yield results that statistically indistinguishable from statistical stemming. The technique had the added advantage of detecting variable spellings of transliterated named entities.

For future work, we would like to try the proposed technique on other languages, because it would likely be effective in automatically learning character-level morphological transformations as well as overcoming some of the problems associated with stemming. It is worthwhile to devise models that concurrently generate morphological and phonologically related tokens.

References

M. A. Ahmed. (2000). A Large-Scale Computational Processor of the Arabic Morphology, and Applications. A Master's Thesis, Faculty of Engineering, Cairo University, Cairo, Egypt.

M. Aljlayl, S. Beitzel, E. Jensen, A. Chowdhury, D. Holmes, M. Lee, D. Grossman, O. Frieder. IIT at TREC-10. In TREC. 2001. Gaithersburg, MD.

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.

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.

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

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

K. Darwish. (2002). Building a Shallow Morphological Analyzer in One Day. ACL Workshop on Computational Approaches to Semitic Languages. 2002.

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.

K. Darwish, D. Oard. (2002). Term Selection for Searching Printed Arabic. SIGIR, 2002, p. 261 - 268.

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.

A. El-Kahki, K. Darwish, M. Abdul-Wahab, A. Taei (2012). Transliteration Mining Using Large Training and Test Sets. NAACL-2012.

F. Gey, D. Oard (2001). The TREC-2001 Cross-Language Information Retrieval Track: Searching Arabic Using English, French or Arabic Queries. TREC, 2001. Gaithersburg, MD. p. 16-23.

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

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.

C. Jacquemin (1997). Guessing morphology from terms and corpora. ACM SIGIR-1997, p.156-165.

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

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, ACL-2007, demonstration session, Prague, Czech Republic, June 2007.

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.

Y. Lee, K. Papineni, S. Roukos, O. Emam, H. Has-san (2003). Language Model Based Arabic Word Segmentation. ACL-2003, p. 399 - 406.

J. Mayfield, P. McNamee, C. Costello, C. Piatko, A. Banerjee. JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. In TREC 2001. Gaithersburg, MD. p. 322-329.

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.

D. Oard, F. Gey (2002). The TREC 2002 Arabic/English CLIR Track. TREC-2002.

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

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

R. Udupa, K. Saravanan, A. Bakalov, A. Bhole. 2009. "They Are Out There, If You Know Where to Look": Mining Transliterations of OOV Query Terms for Cross-Language Information Retrieval. ECIR-2009, Toulouse, France, 2009.

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

J. Xu, A. Fraser, and R. Weischedel (2001). 2001 Cross-Lingual Retrieval at BBN. TREC 2001, pp. 68 - 75.