Answering English Queries in Automatically Transcribed Arabic Speech

Abdusalam F.A. Nwesri S.M.M. Tahaghoghi Falk Scholer
School of Computer Science and Information Technology
RMIT University, GPO Box 2476V, Melbourne 3001, Australia
{nwesri,saied,fscholer}@cs.rmit.edu.au

Abstract

There are several well-known approaches to parsing Arabic text in preparation for indexing and retrieval. Techniques such as stemming and stopping have been shown to improve search results on written newswire dispatches, but few comparisons are available on other data sources. In this paper, we apply several alternative stemming and stopping approaches to Arabic text automatically extracted from the audio soundtrack of news video footage, and compare these with approaches that rely on machine translation of the underlying text. Using the TRECVID video collection and queries, we show that normalisation, stopword-removal, and light stemming increase retrieval precision, but that heavy stemming and trigrams have a negative effect. We also show that the choice of machine translation engine plays a major role in retrieval effectiveness.

Keywords Arabic information retrieval, Cross-language information retrieval, Machine translation.

1 Introduction

Arabic is the official language of twenty-six countries, and is widely spoken in the Islamic world. Information retrieval from Arabic text has attracted increased attention due to its widespread use and the raised interest in Arabic speakers by the media and intelligence communities. In particular, analysis of text automatically extracted from spoken Arabic — whether in broadcast media or monitored conversations — and automatically translated into English, is the subject of intense research.

In this work, we present a comparison of several text retrieval approaches when applied to text extracted from an automatic speech recognition (ASR) algorithm applied to video footage of Arabic television news.

The typical approach to retrieve video using Arabic text is to generate ASR scripts and automatically translate this text to English; this text is then searched using English queries [15]. To the best of our knowledge, retrieving video using the original Arabic ASR text and translated English queries has not been investigated. Most research on Arabic information retrieval has been performed using the text of newswire dispatches [16]; this is substantially different from the noisy text that is obtained from automatically extracted speech. We investigate using such noisy data, and also assess the effectiveness of machine translation of the English queries and the Arabic documents.

The remainder of this paper is organised as follows. In Section 2, we describe the principal techniques used to prepare Arabic text for indexing and retrieval. In Section 3, we describe the collections and algorithms that we use in the experiments described in Section 4. We discuss our results and findings in Section 5, and conclude the paper with our thoughts for future work in Section 6.

2 Arabic Retrieval Techniques

Arabic is a root-based language with a high inflection rate. Words are generated from root words by adding prefixes, suffixes and infixes. To effectively search Arabic text, words that share the same stem or root must be conflated to their roots by removing affixes [13]. There are several principal techniques used to improve the effectiveness of Arabic text retrieval [1]. These include normalising the text by dealing with common typographical conventions, removing highly frequent words (stopwords) such as conjunctions and prepositions, removing prefixes and suffixes through stemming, morphological analysis, and root extraction. We continue with a description of several important techniques.

2.1 Normalisation

In normalisation, different Arabic typographical styles are mapped to a single consistent style. The Arabic alphabet has twenty-eight characters, but lacks vowels; instead, eight diacritics are used above or below letters to indicate the way each written character should be inflected. With the exception of text written for children or for learners of the language, Arabic text is written almost entirely without

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

these diacritics. Even so, certain letters appear with or without a diacritic depending on the preference of the writer. For example, the letter "ا" (*Alef*) may also be written as "أ", "إ", or "آ"; "ي" (*YAA*) as "ى"; and "ة" (*Ta Marbutah*) as "ه". This results in different spellings for words that incorporate these characters.

Characters are attached to each other with no ligatures; the shape of some characters varies depending on whether they appear at the beginning, middle, or end of a word. Modern computer character sets rely on text being post-processed to select the correct shape of a letter.

Finally, the distance between characters can be elongated by using the *Tatweel* or *Maad* character, ـ; for example, سارا <the name "Sarah"> may be written as سـارا. This calligraphic construct is widely used — either manually or automatically — in fully justified text.

To cater for such varying text, normalisation is used as part of the parsing process. Well-known normalisation techniques for Arabic include [5, 10, 11]:

- Removing diacritics and Tatweel characters
- Normalising the letters أ, إ, آ to ا
- Normalising the final ى to ي
- Normalising the final ة to ه

Normalisation should not be applied to foreign or *out of vocabulary* words [14].

2.2 Stopping

Words that appear very frequently in a document collection are considered to add little document-specific information. To avoid the noise that is likely to arise from such generic terms, as well as to reduce the size of the index, they are often omitted during the indexing stage [4].

Stopword lists drawn up for Arabic [3, 10] contain well-known pronouns, prepositions and function words. However, the lists also differ substantially, and no single widely accepted list exists. Importantly, most lists include a single version of each word, despite the fact that Arabic words have different forms. For example, the word في (*in*) is a stopword in almost all Arabic information systems, this word occurs in many other forms such as فيه (*in it*), فيها (*in it* feminine), فيهما (*in them* dual), and so on.

Abu El-Khair [6] studied this approach and proposed three lists; a general stopword list containing 1 377 words, a corpus-based stoplist with 235 words, and a combination of the previous two with 1 529 words. Chen and Gey [5] describe a stoplist created by translating 541 681 unique Arabic words to English and then capturing all words that translate to an English stopword. Their list had 3 447 words.

Despite this disagreement on the stoplist size and content, there is an agreement that removing them from Arabic text improves retrieval precision.

2.3 Stemming

The main objective of stemming is to conflate closely related co-derivative words; this can help improve recall while reducing the number of distinct index entries. There are two different stemming techniques used for Arabic:

Light Stemming strips out specified affixes from the beginning and end of a word, but ignores infixes.

The exact affixes removed vary between stemmers [2, 11, 5, 9]. In this paper, we test the performance of the well-known Larkey light stemmer.

Heavy Stemming aims to return words to their roots by first applying light stemming to remove prefixes and suffixes, and then identifying the appropriate root by matching against well-known patterns.

The Khoja heavy stemmer [10] is widely used in the literature. This stemmer strips out known prefixes and suffixes and returns the root by matching the remaining stem with particular patterns and roots.

The performance of each approach is highly dependent on the list of affixes or roots used. Both light stemming [2, 11] and heavy stemming [11, 12] have been reported to be effective. However, it is unclear whether the reported results apply to noisy data such as the automatic speech recognition (ASR) text that is the subject of this paper.

2.4 Tokenisation

In this technique, the text is split into overlapping tokens of size n, and it is these tokens that are indexed [18]. At search time, queries are similarly tokenised for index lookup. This technique is fast, language independent, and robust against spelling mistakes.

Using six Arabic corpora, AlShehri [3] found the optimal size of overlapping tokens to be three. Xu et al. [18] show that stem-based tokens give better retrieval results than the word-based tokens, and also conclude that tokens of size three are the best option for Arabic. The language independence of this technique and its robustness against noisy and misspelt text makes it promising for our target application.

3 Resources

We now describe collections, translation tools, and stemming algorithms we use in our experiments.

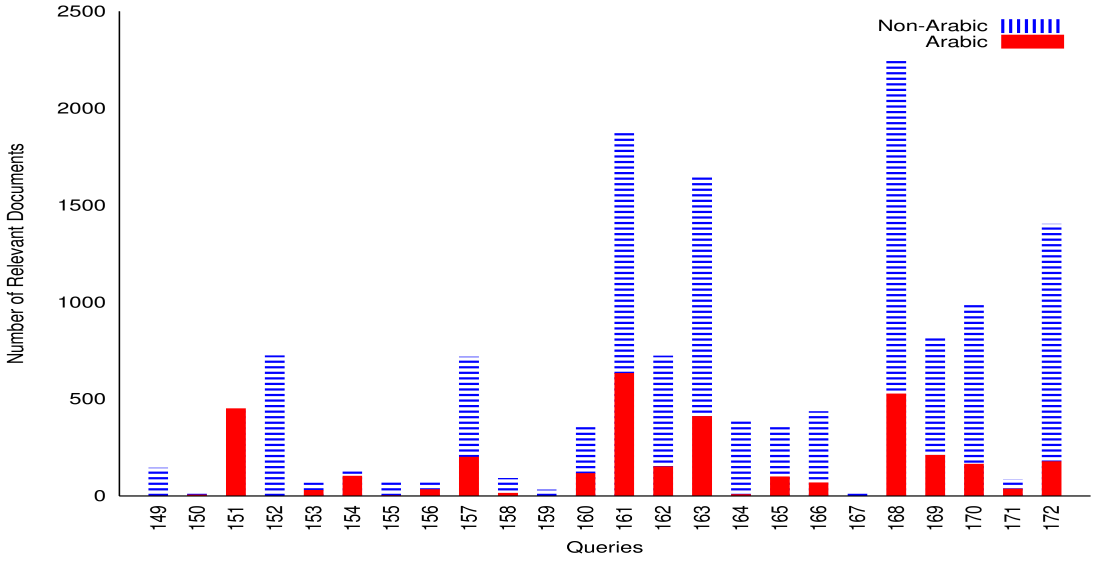

Figure 1. The number of documents relevant to each query for the Arabic and non-Arabic documents in the collection

3.1 Collection Description

The TRECVID 2005 data set contains recorded television broadcast news in three languages — Arabic, Chinese, and English — with the associated ASR transcripts available [15]. Of the total of 169 hours of footage, 43 hours are in Arabic, 52 hours are in Chinese, and 74 hours are in US English. The collection has 24 English-language queries to be used to find specific video footage in the entire collection. The queries all begin with the phrase, "find the shot of", and aim to find scenes containing a specific person, place or object, or a general view, building, or action.

The TRECVID ground truth for measurement of retrieval performance is prepared by manually identifying video shots — sections of video footage that correspond to a single camera operation — that satisfy the information need of the user based on the *visual content*.

To create a text-based test set, we aligned the ASR text with the corresponding shots in the video stream. To allow for speed variations and gaps in speech, the text for each shot is the ASR text that temporally corresponds with that shot and the two shots on either side. The text corresponding to each shot is considered an independent document that is then indexed using a text search engine. A reasonable alternative would be to use story-aligned text [7, 8], rather than shot-aligned text, as the unit of retrieval; we do not explore story alignment in this work.

We interpret the relevance judgments in the context of this alignment; a document is relevant to the query if its corresponding shot has been indicated as being relevant in the ground truth.

For the work described in this paper, we focus on only the Arabic data, comprising 26% of the entire TRECVID 2005 collection. The distribution of relevant documents shows that of 13 945 relevant documents, only 3 475 are Arabic. Similarly, the collection-wide average number of relevant documents per query is 581.0, while for the Arabic subset, the average number of relevant documents per query is 144.8. Figure 1 shows the number of relevant Arabic and non-Arabic documents for each query. Naturally, the smaller pool of relevant answers will lead to lower retrieval performance than that reported for work that uses the entire collection. Since we use only the Arabic text, we extracted the relevance judgements for only the Arabic documents in the pool.

3.2 Automatic Translation Tools

To evaluate the English queries against the Arabic text, we use three different online automatic translation tools to render the queries into Arabic. These are *AlMisbar*,[1] *Google Translate*,[2] and *Systran*.[3] We expect that the choice of translation tool can affect the quality of the translation, and hence the retrieval effectiveness.

[1] http://www.almisbar.com

[2] http://translate.google.com

[3] http://www.systransoft.com

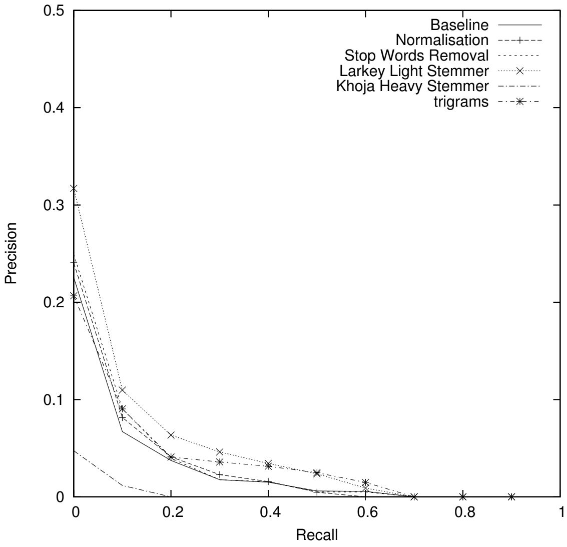

Figure 2. Precision, queries translated with AlMisbar

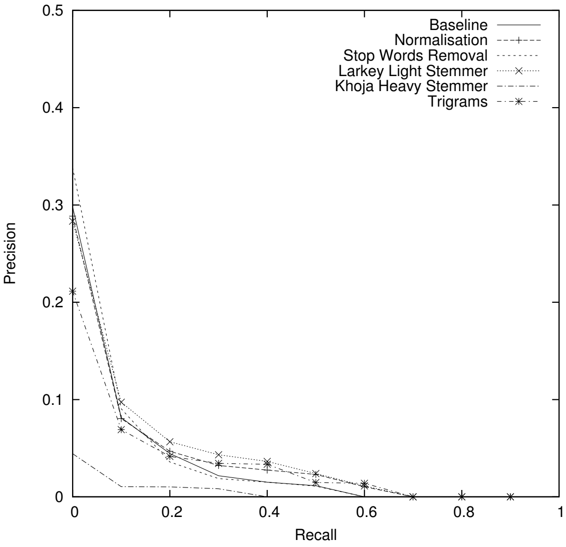

Figure 3. Precision, queries translated with Google Translate

3.3 Stemmers and Retrieval Engines

We used the Lemur toolkit[4] to index the collection, and to evaluate the queries against the collection. Lemur incorporates a built-in Arabic stemmer [11] that supports most of the techniques we have described for Arabic search: normalisation, stopword removal, and light stemming.

We have separately implemented the Khoja stemmer [10] to test the effectiveness of heavy stemming on this collection. This stemmer removes prefixes and suffixes, and checks for pattern matches after each affix removal; it extracts and returns the root if a match is found in the root-word dictionary, and returns the original word otherwise.

To test tokenisation, we chose to use n-grams of size three, which have been reported to produce good results for Arabic retrieval [18].

4 Experiments

To evaluate the effectiveness of different techniques used in Arabic Information retrieval, we designed five different runs using the translated queries:

1. normalise the queries and run them against the normalised ASR text;

2. stop the queries and run them against the ASR text;

3. stem the queries using the Larkey light stemmer, and run them against the similarly-stemmed ASR text;

	Machine Translation		
	AlMisbar	Google	Systran
Baseline	0.0669	0.0686	0.0317
Normalisation	0.0752	0.0752	0.0406
Stopword removal	0.0746	0.0705	0.0326
Larkey Light Stemmer	0.0811	0.0653	0.0479
Khoja Root Stemmer	0.0033	0.0046	0.0048
Trigrams	0.0527	0.0409	0.0318

Table 1. Mean average precision of the techniques

4. stem the queries using the Khoja heavy stemmer, and run them against the similarly-stemmed ASR text; and

5. tokenise the queries into 3-grams and run them against the similarly-tokenised ASR text.

We evaluate retrieval effectiveness using the standard information retrieval measures of Precision (the proportion of retrieved documents that are relevant), and Recall (the proportion of all relevant documents that are retrieved) [17]. As a baseline for comparison, we run the translated queries directly against the ASR text.

5 Results and Discussion

The results for each run are shown in Table 1. The techniques have a clear impact on retrieval performance: with

[4] http://www.lemurproject.org

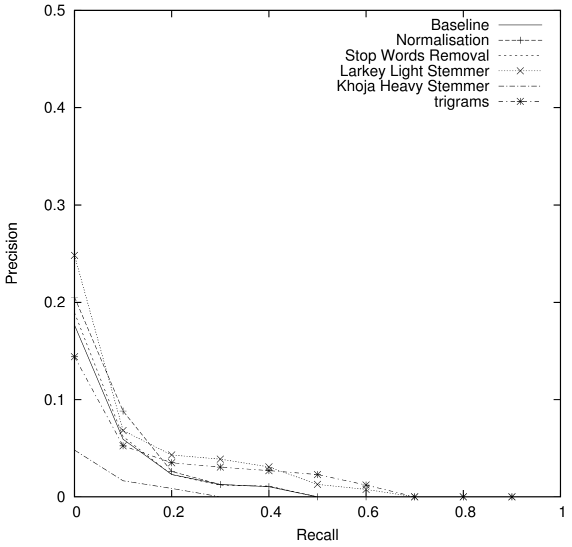

Figure 4. Precision, queries translated with Systran

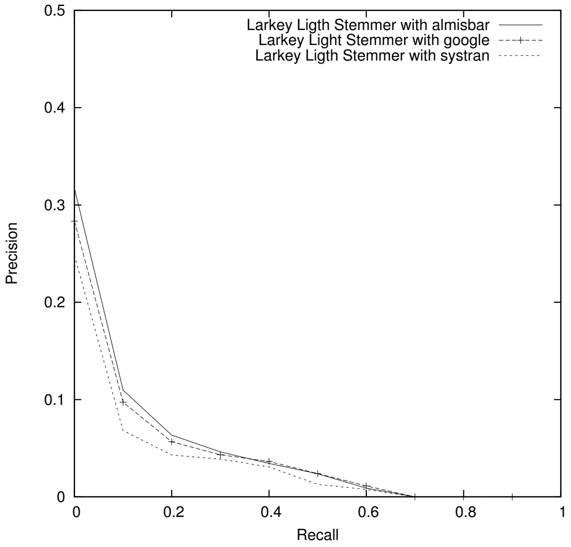

Figure 5. Results for the Larkey light stemmer across translation systems

the exception of heavy stemming and trigrams, all produce improved performance over the baseline. As can be seen from Figures 2, 3, and 4, this improvement is consistent across all three translation systems.

This is an important finding, confirming that the approaches are useful even for noisy data such as that used in these experiments. Light stemming appears to produce the most improvement, followed by stopword removal, and then normalisation. Surprisingly, trigrams performed poorer than the baseline.

In contrast to previous reported results [11], heavy or root stemming actually leads to poor results on this noisy data.

It is also clear that the choice of machine translation has great impact on the results. For instance, the best precision result achieved using Systran is 0.0479 which is below the baseline result for Google and AlMisbar Translates. Overall, AlMisbar is the best of the three translation systems; it produces the highest precision when using light stemming. Figure 5 shows the impact of the translation system choice on retrieval performance when applying light stemming.

We observed that root stemming is the only technique that is significantly worse than the baseline when using Google (Wilcoxon signed rank test, p=0.004975). Results produced by the AlMisbar translation system with light stemming are better than the baseline, but the difference is only weakly significant (p=0.07); this is also the case when applying normalisation and using the Systran system (p=0.05). We note that it is difficult to achieve significant differences based on the relatively small number (24) of available queries.

No automatic translation system is perfect, and, as expected, all three of the translation tools we used had difficulty in finding correct Arabic equivalents for some of the English words in the queries. For instance, the word "court" appears in two English queries, both times in the sense of an open space for games. None of the translation systems produced the correct Arabic meaning "ملعب"; instead they all translated it to محكمة <law court> despite the fact that queries also contained the word "player".

AlMisbar and Google were both successful in translating most proper nouns in the queries, while Systran transliterated such words inconsistently. Surprisingly, all three translation systems failed to translate the proper noun "Baghdad" to its Arabic equivalent. In addition, Google Translate frequently incorrectly spells words containing the Hamza character ء; for example, the word "airplane" is translated to the correct meaning but with the incorrect spelling طاءرة rather than طائرة.

The noisy data produced by the ASR subsystem is another source of errors, with proper nouns frequently transcribed incorrectly. For instance, the name "Condoleeza Rice" is transcribed completely into one word "كوندوليسارايس" instead of two separate ones. Since the AlMisbar and Google Translate systems translate the English name into the correct Arabic equivalent, no match will be found for these terms in the search engine index.

Apart from the use of noisy ASR data and machine translations, our experiments depart from typical information retrieval research in that the underlying relevance assessments are based on the visual content of the shots, and not on the spoken text. Thus, while the comparison of approaches is correct, our absolute results are not directly comparable with other work on Arabic text retrieval. However, the results are comparable to other retrieval results undertaken as part of TRECVID, with the qualification that we use only the Arabic subset of the entire collection of Arabic, Chinese, and English ASR data.

6 Conclusions

Retrieval of information from Arabic text is complicated by the morphology of the language and the large variations that occur in its usage. In this work, we have evaluated the effect of several preprocessing and translation approaches for a noise data set of Arabic text. Our results show that stopping, light stemming, and tokenisation improve retrieval effectiveness, but that heavy stemming and trigrams have a negative impact. We have also shown that the choice of the machine translation engine has a large impact on measured performance in such experiments. As part of our ongoing participation in TRECVID, we plan to augment our collection with the TRECVID 2006 test data and queries; we also plan to use stories, rather than overlapping windows of shot-aligned text, as the unit of retrieval, and to incorporate Arabic search as part of a multi-lingual retrieval system for video.

References

[1] I. A. Al-Sughaiyer and I. A. Al-Kharashi. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213, 2004.

[2] M. Aljlayl and O. Frieder. On Arabic search: improving the retrieval effectiveness via a light stemming approach. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 340–347, McLean, Virginia, USA, 2002. ACM Press.

[3] A. M. AlShehri. *Optimization and effectiveness of n-grams approach for indexing and retrieval in Arabic information retrieval systems*. PhD thesis, School of Information Science, University of Pittsburgh, 2002.

[4] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[5] A. Chen and F. Gey. Building an Arabic stemmer for information retrieval. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*. National Institute of Standards and Technology, 2002.

[6] I. H. A. El-Khair. *Effectiveness of document processing techniques for Arabic information retrieval*. PhD thesis, School of Information Science, University of Pittsburgh, 2003.

[7] W. H. Hsu and S.-F. Chang. Visual cue cluster construction via information bottleneck principle and kernel density estimation. In *Proceedings of the 4th International Conference on Image and Video Retrieval (CIVR)*, Singapore, July 20-22 2005.

[8] W. H. Hsu, L. Kennedy, S.-F. Chang, M. Franz, and J. Smith. Columbia-IBM news video story segmentation in TRECVID 2004. Technical report, Columbia ADVENT, New York, 2005. 209-2005-3.

[9] Kareem Darwish and Douglas W. Oard. CLIR Experiments at Maryland for TREC-2002: Evidence Combination for Arabic-English Retrieval. Technical Report LAMP-TR-101,CS-TR-4456,UMIACS-TR-2003-26, University of Maryland, College Park, February 2003.

[10] S. Khoja and R. Garside. Stemming Arabic text. Technical report, Computing Department, Lancaster University, Lancaster, 1999.

[11] L. S. Larkey, L. Ballesteros, and M. E. Connell. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 275–282, Tampere, Finland, 2002. ACM Press.

[12] L. S. Larkey and M. E. Connell. Structured queries, language modeling, and relevance modeling in cross-language information retrieval. *Information Processing and Management Special Issue on Cross Language Information Retrieval*, 41(3):457–473, 2005.

[13] A. F. A. Nwesri, S. M. M. Tahaghoghi, and F. Scholer. Stemming Arabic conjunctions and prepositions. In M. Consens and G. Navarro, editors, *Lecture Notes in Computer Science: 3772 - Proceedings of the International Symposium on String Processing and Information Retrieval (SPIRE)*, pages 206–217, Buenos Aires, Argentina, 2–4 November 2005. Springer, Heidelberg, Germany.

[14] A. F. A. Nwesri, S. M. M. Tahaghoghi, and F. Scholer. Capturing out-of-vocabulary words in Arabic text. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia, 22–23 July 2006. Association for Computational Linguistics. 258-266.

[15] P. Over, T. Ianeva, W. Kraaij, and A. Smeaton. TRECVID 2005: An overview. In *Proceedings of the TRECVID 2005 Workshop*, Gaithersburg, Maryland, USA, 14–15 November 2006.

[16] E. M. Voorhees and L. P. Buckland, editors. *The Eleventh Text Retrieval Conference, TREC-2002, Gaithersburg, Maryland, USA, November 19-22, 2002, Proceedings*, volume NIST Special Publication:SP 500-251. National Institute of Standards and Technology, 2002.

[17] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers Inc., second edition, 1999.

[18] J. Xu, A. Fraser, and R. Weischedel. Empirical studies in strategies for Arabic retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 269–274, Tampere, Finland, 2002. ACM Press, New York, USA.