ISBN 978-83-60810-22-4

A Web Statistics based Conflation Approach to Improve Arabic Text Retrieval

Farag Ahmed
Data & Knowledge Engineering Group
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Email: farag.ahmed@ovgu.de

Andreas Nürnberger
Data & Knowledge Engineering Group
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Email: andreas.nuernberger@ovgu.de

***Abstract*—We present a language independent approach for conflation that does not depend on predefined rules or prior knowledge of the target language. The proposed unsupervised method is based on an enhancement of the pure n-gram model that is used to group related words based on a revised string-similarity measure. In order to detect and eliminate terms that are created by this process, but that are most likely not relevant for the query (”noisy terms”), an approach based on mutual information scores computed based on web statistical co-occurrences data is proposed. Furthermore, an evaluation of this approach is presented.**

I. Introduction

ARABIC is a Semitic language that is based on the Arabic alphabet containing 28 letters. Its basic feature is that most of its words are built up from, and can be analyzed down to common roots. The exceptions to this rule are common nouns and particles. Arabic is a highly inflectional language with 85% of words derived from triliteral roots. Nouns and verbs are derived from a closed set of around 10,000 roots [1]. Arabic has three genders, feminine, masculine, and neuter; and three numbers, singular, dual, and plural.

The specific characteristics of Arabic morphology make the Arabic language particularly difficult for developing natural language processing methods for Information Retrieval (IR). Arabic is different from English and other Indo-European languages with respect to a number of important aspects: words are written from right to left; it is mainly a consonantal language in its written forms, i.e., it excludes vowels; its two main parts of speech are the verb and the noun in that word order, and these consist, for the main part, of triliteral roots (three consonants forming the basis of noun forms that are derived from them); it is a morphologically complex language, in that it provides flexibility in word formation: as briefly mentioned above, complex rules govern the creation of morphological variations, making it possible to form hundreds of words from one root [2]. Furthermore, the letter shapes are changeable in form, depending on the location of the letter at the beginning, middle or at the end of the word.

One of the main problems we face when indexing and retrieving unstructured text is the variations in word forms. These variations result from the morphological variants, for example in English, verb forms like walk, walked, walking. In the Arabic language, the variations are even more abundant and a word can sometimes be represented by more than 100 different forms. These variation in word forms results from the fact that Arabic nouns and verbs are heavily prefixed. The definite article ال*āl* “the” is always attached to nouns, and many conjunctions and prepositions are also attached as prefixes to nouns and verbs, hindering the retrieval of morphological variants of words. In Table I some word form variations for the word ”student” is presented in order to clarify this issue. The absence of these word form variations in the user query causes a loss of vast amounts of retrieved information. One way to tackle such problems are conflation methods. Conflation is a general term for all processes of merging together nonidentical words that refer to the same principal concept, i.e., merging words that belong to the same meaning class. The primary goal of conflation is to allow matching of different variants of the same word. In natural language processing, conflation is the process of merging or lumping together nonidentical words that refer to the same principal concept. In the context of information retrieval (IR), conflation has a more restricted meaning and usually refers to the process of grouping together morphological variants of the same or related words [3]. Since the variants have similar semantics, it is possible to consider them as equivalent for the purpose of the retrieval tasks. Applying conflations approaches in morphologically complex languages, such as Arabic, improves the retrieval effectiveness and frees the users from taking into account all variants of the same word. Therefore, conflation approaches can be quite beneficial in many fields such as information retrieval and word-processing systems. In order to solve or at least alleviate some of the problems raised by a high inflectional morphology, the stem of the word need to be detected. There are two, widely used, stemming approaches: First, approaches that are language dependent and designed to handle morphological variants. In stemming, morphological variants are reduced to common basic form called root, and second, string-similarity approaches i.e (n-gram), which are (usually) language independent and designed to handle all types of word variants. In this paper, the proposed approach is based on the enhancement of the n-gram pure approach therefore we will focus in describing the n-gram approach in more detail with respect to the Arabic language.

978-83-60810-22-4/$25.00 © 2011 IEEE

TABLE I
WORD FORM VARIATIONS FOR طالب *ṭālb* (STUDENT).

Feminine	Masculine	English
طالبة *ṭālbh*	طالب *ṭālb*	student
الطالبة *ālṭālbh*	الطالب *ālṭālb*	the student
طالبتان *ṭālbtān*	طالبان *ṭālbān*	(two) students(dual)
بطالبة *bṭālbh*	بطالب *bṭālb*	by student
بالطالبة *bālṭālbh*	بالطالب *bālṭālb*	by the student
وطالبة *wṭālbh*	وطالب *wṭālb*	and student
والطالبة *wālṭālbh*	والطالب *wālṭālb*	and the student
للطالبة *lṭālbh*	للطالب *lṭālb*	to the, for a student
طالبته *ṭālbtha*	طالبه *ṭālbh*	his student
طالبتها *ṭālbthā'*	طالبها *ṭālbhā*	her student
طالباته *ṭālbāth*	طلبته *ṭlbth*	his students
طالباتها *ṭālbāthā'*	طلبتها *ṭlbthā*	her students

II. CONFLATION TECHNIQUES

Conflation algorithms can be categorized into four main classes: affix removal, table lookup, successor variety, and n-gram [4]. Affix-removal algorithms reduce a word to its morphological root or a stable stem by stripping off suffixes and prefixes in order to determine the stem. They are the most popular group of conflation algorithms, mainly due to the work of [5] and [6]. In the table-lookup approach, all desired stems for a particular surface-form word are stored in a table. Therefore, this approach can be implemented in a computationally efficient manner, since no word transformation is needed. However, one of the main drawbacks is that due to its manual creation usually not all words and word forms can be covered and thus table-lookup approaches are in most cases domain-dependent. The successor-variety approach was first introduced by Hafer and Weiss (1974) [7]. In successor variety, a lexical text is segmented into stems and affixes. The method uses statistical properties-successor and predecessor variety counts-of a corpus, in order to identify the root [8]. The idea is to count the number of different letters encountered after (or before, respectively), a part of a word and to compare it to the counts before and after that position. Morpheme boundaries are then likely to occur at sudden peaks or increases of that value [9].

In the following, we describe two main approaches (Stemmer and n-gram) which are used to solve or at least alleviate some of the problems raised by a high inflectional morphology.

A. Stemmer Approaches

In information retrieval systems stemming is used to reduce variant word forms to common roots and thereby improve the ability of the system to match query and document vocabulary [10]. Although stemming has been studied mainly for English, stemming approaches have also been developed for several other languages .e.g., Latin [11], Indonesian [12], Swedish [13], Dutch [14], German [15] and Arabic [16]. There are three main types of approaches for stemming, dictionary-based, rule-based, and statistical-based (mainly n-gram based) approaches [17].

Dictionary-based approaches provide very good results at the cost of high development efforts for the dictionary. The dictionary contains all known words with their inflection forms. The main weakness for this approach is the missing words in the dictionary which would not be recognized by the system for stemming. Another weakness is the inability of this method to stem inert names and foreign words. Also the need to process a large dictionary during runtime can result in high requirements for storage space and processing time. The closest Arabic equivalent for this kind of stemmer is the *root-based stemmer* for Arabic [18] which is based on extracting the root of a given Arabic surface word by striping off all attached prefix and/or suffix then attempt to extract the root of it. Several morphological analyzers were developed based on this concept [19], [18]. The weaknesses of this stemmer is that the construction of the corresponding dictionaries or rules is a tedious and labor-intensive task due to the result of the morphology complexity of Arabic language. Another problem is that only some small linguistic resources are available for Arabic language. The second type are the *rule-based approaches*. They are based on set of predefined conditions rules. The most well known stemmer of this type is Porter stemmer [5]. The main weakness for this stemmer is that building the rules for the arbitrary language is time consuming. Furthermore, there is a need for experts with linguistic knowledge in that particular language. The Arabic equivalent for this is the *Light stemmer* [16]. Unlike English, both prefixes and suffixes need to be removed for effective stemming. it is based on striping off prefix and suffix from the word, it uses predefined list of prefix and suffix, it is simply striping off prefix and/or suffix without any further processing in the rest of the stemmed word [20], [16]. The weakness of this stemmer is that the striping off prefixes or suffix in Arabic is a not an easy task. Removing them can lead to unexpected results, as many words start with one letter or more which can mistakenly assumed to be prefix or suffix.

III. n-GRAM

The main idea of n-gram-based approaches, which groups together words that contain identical character substrings of length n, called n-grams, is that the character structure of the word can be used to find semantically similar words and word variants. n-gram, as a conflation technique, differs from stemmers in that they do not require language knowledge, predefined rules, or a vocabulary database. Furthermore, n-gram approaches take into account misspelled and transliterated words [1]. For example, Table II shows 15 different spellings for the name Condoleezza; four of them were found in the same news web site ('CNN-Arabic') [2].

A. n-gram and Arabic Text

Over the last years, several studies, which explore the use of n-grams for processing Arabic text, have been performed.

[1] Transliteration is the process of converting one orthography from one language into another.

[2] http://arabic.cnn.com/ Retrieved on 01/10/2010, www.Google.com

TABLE II
MULTIPLE SPELLINGS FOR THE NAME "CONDOLEEZZA".

S/N	Transliteration	Web Occ.	Comments
1	كونداليزا*kwndālyzā*	3.000.000	CNN
2	كوندوليزا*kwndwlyzā*	197.000	CNN
3	كوندليزا*kwndlyzā*	51.100	CNN
4	كونداليسا*kwndālysā*	26.300	
5	كوندوليسا*kwndwlysā*	26.200	CNN
6	كاندوليزا*kāndwlyzā*	12.700	
7	كنداليزا*kndālyzā*	2.310	
8	كانداليزا*kāndālyzā*	1.530	
9	كونداليزة*kwndālyzh*	491	
10	كندليسا*kndlysā*	344	
11	كونداليزه*kwndālyzh*	195	
12	كنداليسا*kndālysā*	144	
13	كانداليسا*kāndālysā*	9	
14	كونداليسة*kwndālysh*	9	
15	كوندليسي*kwndlysy*	4	

Mayfield et al. (2001) have found that n-grams work well in many languages. Furthermore, they investigated the use of character n-grams for Arabic retrieval in TREC-2001 and found that n-grams of length 4 were most effective [21]. Darwish and Oard examined multiple tokenization strategies for retrieval of scanned Arabic documents. They found that n-grams of size $n = 3$ or $n = 4$ are well suited to Arabic document retrieval [22]. Mustafa (2004) assessed the overall performance of two n-gram techniques that he called conventional and hybrid. In his results, Mustafa pointed out that the hybrid approach outperforms the conventional approach [23]. However, all of the previous studies rely on studying the use of n-gram on the Arabic text based on the following aspects: The effectiveness of n-gram size and assessing the performance of existing n-gram approaches. None of the prior studies attempt to modify the pure n-gram model, so that it also considers language characteristics, while computing the similarity score, in order to improve its performance. Ghaoui et al. (2005) investigated a new morphological class based language model. They used the Morphological rules to derive the different words in a class from their stem. Furthermore, a linear interpolation between the n-gram model and the morphological model has been evaluated. In their experiments they pointed out that morphological class-based model yields poor performance compared to a classical trigram [24]. The performance of the n-gram in Arabic text has been studied by many researchers. For example, Abu-Salem (2004) found that all of the proposed n-gram methods outperform the Word, Stem, and Root index methods [25]. We would like to emphasize again, that none of the prior studies attempted to modify the pure n-gram model, such that it also considers language characteristics while computing the similarity score, in order to improve its performance. All of the previous studies considered only to investigate the performance on Arabic text based on the effectiveness of n-gram size using existing n-gram approaches. Due to the mentioned insufficiencies of the existing approaches, we propose a "revised" n-gram algorithm that makes it possible to handle one-character infixes, prefixes, and suffixes, which are frequent in Arabic. The proposed method obtained superior results on a large newspaper corpus.

B. Computing Similarity Scores Based on n-grams

The n-gram model can be used to compute the similarity between two strings by counting the number of similar n-grams they share. The more similar n-grams between two strings exist, the more similar they are. Based on this idea the similarity coefficient can be derived. The similarity coefficient δ is defined by the following equation:

$$\delta = \frac{|\alpha \bigcap \beta|}{|\alpha \bigcup \beta|} \tag{1}$$

where α and β are the n-gram sets for two words a and b to be compared. $\alpha \bigcap \beta$ denotes the number of similar n-grams in α and β, and $\alpha \bigcup \beta$ denotes the number of unique n-grams in the union of α and β.

IV. THE PROPOSED APPROACH

We successfully used our revised n-gram approach for the conflation task in [26]. However, the revised n-gram approach in some cases expanded the user query with terms not relevant for the query ("noisy terms"). Here in this paper, we propose an approach, based on computed mutual information scores, based on web statistical co-occurrences data, in order to detect and eliminate such noisy terms.

In the following, we describe first in Section IV-A our algorithm based on the enhancement of the n-gram model, in order to expand the user query with relevant terms; then in Section IV-B, the approach to eliminate any potentially generated noisy terms based on mutual information scores computed on corpora or web based co-occurrence statistics is presented. The n-gram based approach assumes two strings are alike based only on a string similarity comparison: the more n-grams existing between two strings, the more similar they are. However, there are many words that are have a very similar text pattern but a quite different meaning. Therefore, we improved our n-gram approach by eliminating such noisy terms that could have been generated. This is done by computing the cohesion score between all revised n-gram generated expanded terms using the mutual information (MI) measure. The term/terms that have a lower MI score than the MI score mean for all expanded terms can be considered as noisy term/terms and thus will be eliminated.

A. Revised n-gram Approach

Arabic nouns and verbs are heavily prefixed and suffixed as described in the first section. As a result, it is possible to have words with different lengths that share the same principal concept. Furthermore, the pure n-gram based approach to compute the similarity coefficient as described above (see Eq. (1)) does not consider the order of the n-grams in the target word [27]. This increases the probability that the matching score between two strings will be higher even though they do not share the same concept. Therefore, we revised the computation of the similarity between words to take these two aspects into account. For simplicity, we describe our algorithm for $n = 2$ (bigrams). However, the approach can be applied for trigrams and n-grams with $n > 3$ as well. We define bigrams of words by their respective position in the word $w_{i,i+(n-1)}$ where i defines the position of the first letter and $i + (n - 1)$ the position of the last letter of the considered n-gram. Thus, the last possible position of an n-gram, in a word, is defined by $j = |w| - n + 1$ where $|w|$ defines the length of the word. In order to deal with the first and second aspect mentioned above, we define a window of n-grams of the target candidate words that should be compared, i.e., while in Eq. (1) all n-grams are compared with each other, we only compare n-grams that are in close proximity to the position of the n-gram in the word to be compared when computing the similarity score.

Overall, the computation of the similarity score S for a given n-gram size n and a given odd-numbered window size m can be defined as follows. Assuming that u is the longer word (if v is longer than u then u and v can be simply exchanged):

$$S_{n,m}(u,v) = \frac{\sum_{i=2}^{|u|-n+1} \sum_{j=\frac{m-1}{2}}^{\frac{m-1}{2}} g(u_{i,i+(n-1)}, v_{i+j,i+j+(n-1)})}{N} \tag{2}$$

where $g(a,b) = \begin{cases} 1 & if \;\; a = b \\ 0 & \text{otherwise.} \end{cases}$ and

$$u_{i,j} = \begin{cases} substring(u,i,j) & if \;\; i \leq j \\ ,,\,,, & \text{otherwise.} \end{cases}$$

Here, u and v are the words to be compared, the nested sum counts the number of n-grams in v that are similar to n-grams in a window the size of m around the same position in word v. N is computed similarly as in Eq. (1).

B. Mutual Information (MI)

Given a query, the set of possible expanded terms using the revised n-gram will be generated; the coherence between the expanded terms is computed based on mutual information (MI). Giving a source of data, Mutual Information (MI) is a measure to calculate the correlation between terms in specific space (corpus or web). MI based approaches have been used often in word sense disambiguation task e.g., [28], [29]. Here in this paper mutual information approach is used to detect the noise term/terms based on its correlation with other terms in web.

Given a query term $q_i = \{t_1, t_2, ..., t_n\}$ and a set of its revised n-gram model generated expanded terms $\{ext_{i,1}, ext_{i,2}, ..., ext_{i,m_i}\}$, where m_i defines the number of extended terms for t_i and $1 \leq i \leq n$. Given the set of $\frac{n(n-1)}{S}$ combinations, where S is the size of each combinations set, then the set of combinations between all expanded terms is defined as $Com_i = \{\{ext_{i,j}, ext_{i,k}\} | 1 \leq j < n, j < k \leq n\}$. The mutual information of each combination set can be computed based on the following equation:

$$MI(q_{t_1}, q_{t_2}) = log_2 \frac{p(q_{t_i}, q_{t_j})}{p(q_{t_i})p(q_{t_j})} \tag{3}$$

where $p(q_{t_i}, q_{t_j})$ being the joint probability of both expanded terms in the combination sets to occur in web. The probability is estimated by the relative frequency of the expanded terms in a given corpus, here the web, i.e., it is estimated by how many times q_{t_i}, q_{t_j} occur together in a (web) document.

C. A Walk Through Example

To illustrate the improvement of the revised n-gram algorithm using the statistical co-occurrences data obtained from web, let us consider the following example.

The user submit the query صحيفة*ṣḥyfh* (Newspaper), the system using the revised n-gram model with similarity threshold of 60% expanded the user query with the following terms: (بصحيفة*bṣḥyfh* ”by Newspaper”, وصحيفة*wṣḥyfh* ”and Newspaper” , للصّحيفة*llṣḥyfh* ”for the Newspaper” , نحيفة*nḥyfh* (”slim” Feminine) and لصحيفة*lṣḥyfh* ”for a Newspaper”. The algorithm starts by generating all possible combinations between the expanded terms where $Com_i = \{\{ext_{i,j}, ext_{i,k}\} | 1 \leq j < 5, j < k \leq 5\}$. After generating all possible combinations between the expansion terms, the mutual information score for each expansion term combination will be calculated based on Eq. (3). Table III illustrates possible expanded term combinations and their mutual information score. As shown in Table III, one of the expanded term combinations included the expanded term نحيفة*nḥyfh* ”slim”. It has the lowest mutual scores (23.793, 23.790,23,165 and 21.314).

As shown in Table IV, the same expanded term has the lowest MI average score (23.015), which is below the MI score mean (25.456), and thus will be classified by the proposed approach as a noisy term and will be eliminated. In contrast, all other expanded terms have an average mutual score, which is above the MI score mean and thus should be correct expanded terms for the user's query.

V. Evaluation

In our evaluation, we compared our Revised n-gram and (revised n-gram + MI) approaches with the pure n-gram and edit distance approaches. We used $n = 2$ (bigrams) to enable retrieval of short words, as well as other word lengths. In order, to gain a certain degree of accuracy, we obtained the

TABLE III
EXPANDED TERM COMBINATIONS AND THEIR MI SCORES.

Expanded Terms Combinations	MI **Score**
(وصحيفة*wṣḥyfh* ,للصّحيفة*llṣḥyfh*) ”and Newspaper, for the Newspaper”	28.651
(للصّحيفة*llṣḥyfh* ,لصحيفة*lṣḥyfh*) ”for the Newspaper, for a Newspaper”	28.075
(بصحيفة*bṣḥyfh* ,لصحيفة*lṣḥyfh*) ”by Newspaper, for a Newspaper”	27.054
(وصحيفة*wṣḥyfh* ,لصحيفة*lṣḥyfh*) ”and Newspaper, for a Newspaper”	27.047
(بصحيفة*bṣḥyfh* ,وصحيفة*wṣḥyfh*) ”by Newspaper,and Newspaper”	26.486
(بصحيفة*bṣḥyfh* ,للصّحيفة*llṣḥyfh*) ”by Newspaper, for the Newspaper”	25.186
(للصّحيفة*llṣḥyfh* ,نحيفة*nḥyfh*) ”for the Newspaper, slim”	23.793
(بصحيفة*bṣḥyfh* ,نحيفة*nḥyfh*) ”by Newspaper, slim”	23.790
(وصحيفة*wṣḥyfh* ,نحيفة*nḥyfh*) ”and Newspaper, slim”	23.165
(نحيفة*nḥyfh* ,لصحيفة*lṣḥyfh*) ”slim, for a Newspaper”	21.314
The MI score mean	25.456

TABLE IV
EXPANDED TERMS AND THEIR AVERAGE MI SCORES.

Expanded Terms	MI **average Score**
(للصّحيفة*llṣḥyfh*) ”for the Newspaper”	26.421
(وصحيفة*wṣḥyfh*) ”and Newspaper”	26.337
(لصحيفة*lṣḥyfh*) ”for a Newspaper”	25.872
(بصحيفة*bṣḥyfh*) ”by Newspaper”	25.629
(نحيفة*nḥyfh*) ”slim”	23.015

statistical co-occurrence data needed for the MI algorithm, from the web, using the yahoo search engine [3].

A. Data Selection

In order to create the lists of expanded terms for each test query, we crawled the Web for articles published on one popular Arabic news Web site (CNN-Arabic)[4] in the period from January 2002 to March 2007. We obtained 5,792 Arabic documents, all of which are abstracts of articles on politics, sports, art, economy, and information science (size 60 MB). The approaches were evaluated against 500 queries that were formulated randomly, ensuring that the length of the query terms vary and short as well as long query terms are included. In order to construct the random queries, the algorithm requires the availability of a lexicon of terms that were extracted from the test data.

B. Comparison of Conflation Approaches and Web Experiment

To evaluate the proposed approaches, we used Precision and Recall. Precision and Recall are measures used to evaluate the performance of information retrieval (IR) approaches. Precision is the number of relevant documents retrieved divided by the total number of documents retrieved. Recall is the number of relevant documents retrieved divided by the total number of existing relevant documents in the data collection that should have been retrieved.

As table VI shows, in the first experiment, we calculated the average precision (based on the randomly selected 500 queries) for the pure trigram, edit distance, revised bigram and (revised bigram + MI) for the similarity thresholds of 60, 65, 70, 75, 80, 85, 90, and 95% (Table VI shows the precision average). The trigram approaches (pure and revised) achieved higher precision than the revised bigram approach but in the same time it achieved lower recall than the revised bigram as it will be shown next in this section. The revised bigram precision was improved by 3.3% using mutual information approach based on statistical data obtained from web.

For the second experiment, we estimated the average recall and F-measure for a sample of 30 queries out of 500. The query terms were selected in the same way as described in Section V-A. For all queries, the number of relevant documents were obtained manually, by selecting all possible word variations (due to this huge manually work we could only provide this data for 30 queries). We obtianed the precision, recall and F-Measure using five conflations approaches pure-trigram, pure-bigram, edit distance, revised-bigram and (revised-bigram + MI). As shown in Table V the revised bigram approach gained a higher F-measure of up to 84% compared to the pure trigram, pure bigram, and edit-distance approaches. These results showthat the revised n-gram has gained an overall higher degree of retrieval performance than the pure n-gram and edit-distance approaches. As Table V shows, the revised bigram and pure trigram approaches have very similar precision, but the pure trigram approach missed many relevant documents and therefore has a lower average recall than the revised bigram approach.

The pure bigram approach has similar recall compared to the revised bigram approach. The pure bigram approach has lower precision since it does not take into account the order of the n-grams and therefore it is possible that many irrelevant

[3] yahoo.com, search performed on 19/02/2011

[4] http://arabic.cnn.com/

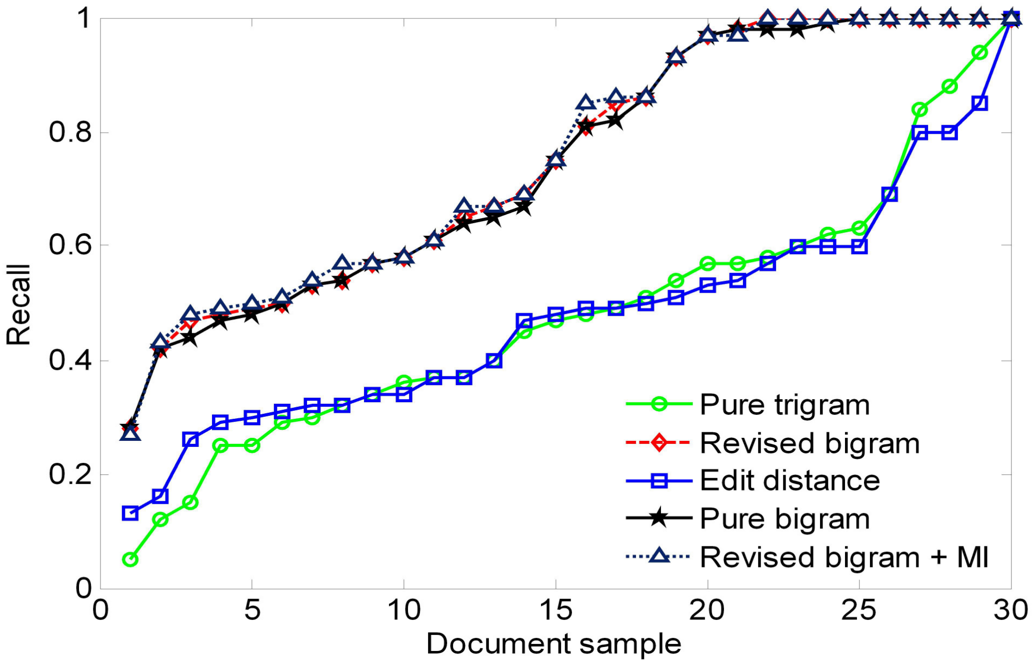

Fig. 1. Average recall for pure trigram, edit distance, pure bigram, revised bigram and (revised bigram +MI) approaches (sorted by recall value).

documents will be retrieved. The approach using edit distance has lower precision. This is due to the fact that this method does not take into account the relationship between characters in the compared words as the n-gram approaches do. Figure 1 illustrates that the revised bigram approach gained a higher average recall than the pure trigram, edit distance and pure bigram approaches, since it took into account different word lengths and similarity enhancement.

For the third experiment, we performed the web experiments using the mutual information approach to improve the precision of revised bigram approach. This was done by eliminating the bigram generated noisy expanded terms as discussed in Section IV-B, Table V and Figure 1 shows that the mutual information approach using statistical co-occurrence data obtained from the web succeeded in eliminating 25 irrelevant expanded terms generated by the revised bigram approach. The failed cases were counted when the algorithm failed to eliminate the noisy terms or when the algorithm eliminate a corrected expanded term/terms along with the noisy one.

For example, we consider the query افريقيا*āfryqyā* ”Africa”, the algorithm succeeded in eliminating the noisy term فريقي*fryqy* ”my team” or ”two teams” but at the same time, it eliminated a relevant term بافريقيا*bāfryqyā* ”by Africa”. One interpretation for this lack, is that the word فريقي*fryqy* ”my team” or ”two teams” with average MI scores (27.999) frequently appeared in the context of African sport and thus it increases the MI score mean (28.437) in that the average MI scores for the relevant word بافريقيا*bāfryqyā* ”by Africa” (27.708) is below the MI score mean.

TABLE VI
AVERAGE PRECISION FOR ALL APPROACHES.

Techniques	**Precision %**
Revised bigram	91.3
Revised-bigram + MI	94.6
Pure bigram	79.4
Revised trigram	98.7
Pure trigram	95.7
Edit distance	87.3

VI. Conclusion

We presented a language-independent conflation approach, i.e., an approach that does not depend on any predefined rules or previous knowledge of linguistic information about the target language. Furthermore, we evaluated our approach succesfully on the Arabic language, which is one of most inflected languages in the world. In order to deal with n-gram nosiy expanded terms, a mutual information appraoch applied to statistical co-occurrences data obtained from web was developed, in that the terms that have less cohesion score with other will be assumed as noisy terms and thus will be eliminated. The eliminations of the n-gram noisy generated terms improved the precision of the revised n-gram with 4%. The failed cases by the algorithm can be interrelated by the lack of the training data or by the very generic term usage where terms can appear in different contexts.

TABLE V

AVERAGE RECALL, PRECISION, AND F-MEASURE FOR THE FIVE APPROACHES FOR A SAMPLE OF 30 QUERIES OUT OF 500.

	Pure-trigram	**Pure-bigram**	**Edit distance**	**Revised-bigram**	**Revised-bigram + MI**
Retrieved	366	629	400	596	571
Relevant	360	539	358	554	554
Irrelevant	6	90	42	42	17
Miss Relevant	6	195	376	180	180
Precision	0.98	0.86	0.89	0.93	0.97
Recall	0.49	0.73	0.49	0.76	0.76
F-Measure	0.65	0.80	0.64	0.84	0.86

REFERENCES

[1] S. Al-Fedaghi and F. Al-Anzi, "A new algorithm to generate arabic root-pattern forms," in *Proceedings of the 11th National Computer Conference, King Fahd University of Petroleum and Minerals*, Dhahran, Saudi Arabia, 1989, pp. 04–07.

[2] H. Moukdad and A. Large, "Information retrieval from full-text arabic databases: Can search engines designed for english do the job?" *International Journal of Libraries and Information Services*, pp. 63–74, 2001.

[3] S. Kosinov, "Evaluation of n-grams conflation approach in text-based information retrieval," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*, 2001, pp. 657–664.

[4] W. B. Frakes, "Stemming algorithms," *Information retrieval: data structures and algorithms*, pp. 131–160, 1992.

[5] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[6] J. Lovins, "Development of a stemming algorithm," *Mechanical Translation and Computational Linguistics*, vol. 11, pp. 22–31, 1968.

[7] H. M. and W. S., "Word segmentation by letter successor varieties," *Information Processing and Management*, vol. 10, pp. 371–386, 1974.

[8] M. Dang and S. Choudri, "Simple unsupervised morphology analysis algorithm," in *Unsupervised Segmentation of Words into Morphemes: Challenge 2005, Laboratory of Computer and Information Science*, 2005.

[9] S. Bordag, "Unsupervised knowledge-free morpheme boundary detection," in *the International Conference on Recent Advances in Natural Language Processing (RANLP 05)*, 2005. [Online]. Available: http://wortschatz.unileipzig.de/?sbordag/papers/BordagMorphy05.pdf

[10] J. Xu and W. B. Croft, "Corpus-based stemming using co-occurrence of word variants," *ACM Transactions on Information Systems*, vol. 16, no. 1, pp. 61–81, 1998.

[11] M. Greengrass, A. M. Robertson, S. Robyn, and Willett, "Processing morphological variants in searches of latin text," *Information research news*, vol. 6, no. 4, pp. 2–5, 1996.

[12] V. Berlian, S. N. Vega, and S. Bressan, "Indexing the indonesian web: Language identification and miscellaneous issues)," in *Proceedings of Tenth International World Wide Web Conference*, Hong Kong, 2001.

[13] J. Carlberger, H. Dalianis, M. Hassel, and O. Knutsson, "Improving precision in information retrieval for swedish us-ing stemming," in *Proceedings of NODALIDA '01 - 13th Nordic conference on computational linguistics*, Uppsala, Sweden, 2001.

[14] W. Kraaij and R. Pohlmann, "Viewing stemming as recall enhancement," in *Proceedings of ACM SIGIR96*, 1996, pp. 40–48.

[15] C. Monz and M. de Rijke, "Shallow morphological analysis in monolingual information retrieval for dutch, german and italian," in *Proc. of Evaluation of Cross-Language Information Retrieval Systems CLEF 2001*, ser. Lecture Notes in Computer Science, vol. 2406. Springer-Verlag, 2002, pp. 262–277.

[16] L. Larkey, L. Ballesteros, and M. Connell, "Light stemming for arabic information retrieval," in *Arabic computational morphology*, A. Soudi, A. V. den Bsch, and G. Neumann, Eds. Netherlands: Springer-Verlag, 2007, vol. 38, pp. 221–243.

[17] A. Gelbukh, M. Alexandrov, and S. Han, *Detecting Inflection Patterns in NL by Minimization of Morphological Model*, ser. LNCS 3287. Springer, 2004, pp. 432–438.

[18] S. Khoja and R. Garside, "Stemming arabic," Website, 1999, available online at http://zeus.cs.pacificu.edu/shereen/research.htm; visited on January 15th 2009.

[19] T. Buckwalter, "Arabic morphological analyzer version 1.0." Website, 2002, available online at http://www.ldc.upenn.edu/; visited on January 8th 2010.

[20] A. N. D. Roeck and W. Al-Fares, "A morphologically sensitive clustering algorithm for identifying arabic roots," in *Proceedings of ACL 2000*, Hong Kong, 2000, pp. 199–206.

[21] J. Mayfield, P. McNamee, C. Costello, C. Piatko, and A. Banerjee, "Experiments in filtering and in arabic, video, and web retrieval," in *Proc. of the Eighth International Symposium on String Processing and Information Retrieval (SPIRE 2001)*, 2001, pp. 136–142.

[22] K. Darwish and D. W. Oard, "Term selection for searching printed arabic," in *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 2002, pp. 261–268.

[23] S. H. Mustafa, "Character contiguity in n-gram-based word matching: the case for arabic text searching," *Processing and Management*, vol. 41, no. 4, pp. 819–827, 2004.

[24] A. Ghaoui, F. Yvon, C. Mokbel, and G. Chollet, "On the use of morphological constraints in n-gram statistical language model," in *Proc. of Interspeech-2005*, 2005, pp. 1281–1284.

[25] H. Abu-Salem, "Comparison of stemming and n-gram matching for term-conflation in arabic text," *International Journal of Computer Processing of Oriental languages*, vol. 17, no. 2, pp. 61–81, 2004.

[26] F. Ahmed and A. Nürnberger, "Evaluation of n-gram conflation approaches for arabic text retrieval," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 60, no. 7, pp. 1448–1465, 2009.

[27] B.-O. Khaltar, A. Fujii, and T. Ishikawa, "Extracting loanwords from mongolian corpora and producing a japanese-mongolian bilingual dictionary," in *Proc. of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*, 2006, pp. 657–664.

[28] F. Ahmed and A. Nürnberger, "multi searcher: can we support people to get information from text they can't read or understand?" in *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2010, pp. 837–838.

[29] F. Ahmed, A. Nürnberger, and M. Nitsche, "Supporting arabic cross-lingual retrieval using contextual information," in *Multidisciplinary Information Retrieval*, A. Rauber and A. de Vries (Eds.), Eds. Berlin-Heidelberg: Springer-Verlag, 2011, vol. 6653, pp. 30–45.