

# N-Grams Conflation Approach for Arabic Text

Farag Ahmed  
Information Retrieval Group  
Faculty of Computer Science  
Otto-von-Guericke-University of Magdeburg  
Tel. +49.391.67-11399  
fahmed@iws.cs.uni-magdeburg.de

Andreas Nürnberger  
Information Retrieval Group  
Faculty of Computer Science  
Otto-n-Guericke-University of Magdeburg  
Tel. +49.391.67-18487  
nuernb@iws.cs.uni-magdeburg.de

## ABSTRACT

In this paper we present a language independent approach for conflation that does not depend on predefined rules or prior knowledge in the target language. Different from prior studies on Arabic text that use pure n-gram models without any attempt for further enhancement on the basis of refined n-gram similarity measures or stemmer techniques which are language-specific, we propose an unsupervised method based on an enhancement of the pure n-gram model that can group related words based on various string-similarity measures. The proposed approach is based on the enhancement of n-gram comparisons that restrict the search to be in specific locations of the target word by taking into account the order of n-grams. We show that the proposed method is effective to achieve high score similarities between all of the word form variations. Furthermore, it reduces the ambiguity, i.e. obtains a higher precision and recall, compared to the pure n-gram based approaches.

## Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]:H.3.3 Information Storage and Retrieval: Information Retrieval and Search—Conflation techniques;

## General Terms

Algorithms, Measurement, Performance, Experimentation, Languages, Verification.

## Keywords

Information retrieval, N-gram approaches, Stemming, Arabic language.

## 1. INTRODUCTION

Conflation is a general term for all processes of merging together nonidentical words which refer to the same principal concept i.e. to merge words which belong to 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 proc-

ess of merging or lumping together nonidentical words which refer to the same principal concept [1]. In the context of information retrieval (IR) conflation has a more restricted meaning and usually refers to grouping together morphological variants of the same or related words [2]. Conflation algorithms can be broadly divided into two main classes: stemming algorithms, which are language dependent and which are designed to handle morphological variants, and string-similarity algorithms, which are (usually) language independent and which are designed to handle all types of variant [3].

### 1.1 Arabic language

Arabic is a Semitic language, it consist of 28 letters, and its basic feature is that most of its words are built up form, 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 tri-lateral roots. Nouns and verbs are derived from a closed set of around 10,000 roots [4]. Arabic has three genders, feminine masculine and neuter; three numbers, singular, dual (represent 2 things), and plural. May be replace by “The specific characteristics of Arabic morphology make Arabic language particularly difficult for developing natural language processing methods for information retrieval. One of the main problems in retrieving Arabic language text is the variation in word forms, for example the Arabic word “kateb” (*author*) is built up from the root “ktb” (*write*). Prefixes and suffixes can be added to the words that have been built up from roots to add number or gender, for example adding the Arabic suffix ”ان“ (*an*) to the word “kateb“ (*author*) will lead to the word “kateban” (*authors*) which represent dual masculine. What makes Arabic complicated to process is that Arabic nouns and verbs are heavily prefixed. The definite article ”ال“ (*al*) 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 [5]. In Table 1 an example for the word *student* is presented in order to clarify this issue. 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 motivated above, complex rules govern the creation of morphological variations, making it possible to form hundreds of words from one root [6]. Furthermore the letters shapes are changeable

**Table 1. Word form variations that share the same principal concept whose English translation contain the word student or students**

English Translation	Feminine	Masculine
Student	طالبة	طالب
The student	الطالبة	الطالب
(two) students(dual)	طالبتان	طالبان
by the student	بالبطالبة	بالبطال
and by the student	والبطالبة	والبطال
By student	بطالب	بطالب
And By student	وبطالبة	وبطال
and my student	وطالبتى	وطالبي
my student	طالبتى	طالبي
as, like student	كالطالبة	كالطالب
to the, for the student	للطالبة	للطالب
so , then , and student	فالبطالبة	فالبطال
to her/his student	لطالبتها	لطالبه
and to the student, and for the student	و للطالبة	و للطال
his student	طالبتة	طالبه
her student	طالبتها	طالبيها
and his student	وطالبتة	وطالبه
and her student	وطالبتها	وطالبيها
their student	طالبتهم	طالبيهم
and her students (Dual)	وطالبتيهما	وطالبيهما
his students	طالباته	طالبيه
her students	طالباتها	طالبيها
and his students	وطالباته	وطالبيه
and her students	وطالباتها	وطالبيها
his students (for 2 persons)	طالبتيهما	طالبيهما
her students (Dual)	طالبتيهما	طالبيهما
their students	طالبتهم	طالبيهم
and their students	وطالبتهم	وطالبيهم
our students	طالباتنا	طالبتنا
and Our students	وطالباتنا	وطالبتنا
his students (Dual)	طالبتيه	طالبيه
and his students (Dual)	وطالبتيه	وطالبيه
By students	بطلبات	بطلبة
And By students	وبطالبات	وبطالبة
More than two(plural) students	طالبات	طلبة
and her students (Dual)	وطالبتيهما	وطالبيهما

in form, depending on the location of the letter at beginning, middle or at the end of the word.

Based on these properties of Arabic language, i.e. that nouns and verbs are massively prefixed and suffixed, we derived the need for modifications of the commonly used n-gram based conflation techniques so that these specific properties are considered. Furthermore, the ambiguity with respect to the similarity score measure of the pure n-gram approach should be reduced.

The remainder of this paper is organized as follows. In Sec. 2 we discuss previous related work on conflation techniques. In Sect. 3 the proposed algorithm is described. The used data, the evaluation and results are discussed in Sect. 4. Some concluding remarks are finally given in Sect. 5.

## 2. Conflation techniques

In the following we briefly discuss the two major conflation techniques: stemmers and n-gram based techniques.

### 2.1 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 [7]. Although stemming has been studied mainly for English, stemming techniques have also been developed for several other languages such as Malay [8], Latin [9], Indonesian [10], Swedish [11] Dutch[12], German [13], French [14], Slovene [15], Turkish [3] and Arabic [16,17]. There are three main approaches for stemming, Dictionary-based, Rule-based, and Statistical-based approaches [18].

*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* which is based on extracting the root of a given Arabic surface word by stripping off all attached prefix and/or suffix then attempt to extract the root of a given Arabic surface word. Several morphological analyzers were developed based on this concept [19] [16]. The weaknesses for this stemmer are: it does nothing when it comes across some words which have no root, for example the Arabic words "نحن" (we), "بعد" (after), "تحت" (under). Furthermore, the construction of the corresponding dictionaries or rules is a tedious and labor consuming 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 approach is the *Rule-Based approach*; it is based on set of predefined conditions rules. The most well known stemmer is Porter stemmer [20]. 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*. Unlike English, both prefixes and suffixes need to be removed for effective stemming. it is based on stripping of prefix and suffix from the word, it use predefined list of prefix and suffix, it is simply stripping of prefix and/or suffix without any further processing in the rest of the stemmed word [21, 17, 22]. The weakness of this stemmer is that the stripping of 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. Due to the fact that all light stemmers use the normalization, which consist of several steps, one of them is to Replace ا and ا and ا with bare alef ا to avoid the ambiguity as most of the Arabic users use just the bare alef ا in their search, this is will lead to the result that all "ال" (al) will be mistakenly identified as prefix even if they are in reality not. Example for that the Arabic words "آلات" (Machines), "آلاف" (Thousands), "آلام" (Afflictions), "آن" (now), "آلم" (pain), "آليات" (Mechanisms). When stripping off all "ال" (al) then the result of the stemmer will be whether other Arabic words, example for that the Arabic word "آلام" when stripping off the "ال" then the result will be "أم" which mean mother, or the result will be not an Arabic word.

## 2.2 N-gram conflation techniques

The main idea of n-gram based approaches, which groups together words that contain identical character sub-strings of length n called n-grams [23], is that the character structure of the word can be used to find semantically similar words and word variants. N-gram as conflation technique differs from stemmers in terms of not requiring language knowledge, predefined rules or a vocabulary database. Furthermore; n-gram approaches take into account the misspelled and the transliterated words.

### 2.2.1 N-Gram and Arabic text

Over the last years there were several studies which explore the use of n-grams for processing Arabic text. Mayfield et al. [24] 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. Darwish and Oard examined multiple tokenization strategies for retrieval of scanned Arabic documents, they found out that n-grams of size n=3 or n=4 are well suited to Arabic document retrieval [25]. In [26] Suleiman H. Mustafa assessed the overall performance of two n-gram techniques that he called conventional and hybrid. The conventional approach combines as usual for comparison the first character with the second and second with third and so on till  $w_{n-1} + w_n$ . The so-called hybrid approach combines the first character with the second and first with third then second with third and second with fourth till  $w_{n-2} + w_{n-1}$ ,  $w_{n-2} + w_n$ ,  $w_{n-1} + w_n$ . Furthermore, three different levels of word stemming were applied: no stemming, light stemming, and higher-order stemming. In his results Mustafa pointed out that the hybrid approach outperforms the conventional approach. Classifying Arabic text using n-gram frequencies also have been fruitful [27]. However, all of the previous studies rely on the investigation of the use of n-gram on the Arabic text based on those factors: 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 such that it considers also language characteristic while computing the similarity score in order to improve its performance.

## 3. 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  $\delta$  is defined by the following equation:

$$\delta_n(a, b) = \frac{|\alpha \cap \beta|}{|\alpha \cup \beta|} \quad (1)$$

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

### 3.1 Revised n-gram approach

Arabic nouns and verbs are prefixed and suffixed as described in the first section. As a result of that, it is possible to have words with different lengths that share same principal con-

cept. Figure 1 shows an example of two Arabic words: استمرارية (Continousness) and استمرار (Continued) that have different length but belong to same meaning class.



Figure 1 Bigram similarity measure between 2 words with different lengths

Furthermore, the pure n-gram based approach to compute the similarity coefficient as described above Eq (1), does not consider the order of the n-grams in the target word [28]. 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 a similarity between words to take these two aspects into account.

Based on our previous work [29] where we applied a revised n-gram approach (Multispell) for spelling error corrections, we propose here a modified version for the conflation task. 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. For example, for a window of size 3, which is the average of the Arabic prefix length, the search will shift to the left or right side. An example is given in Fig. 1, where  $w'$  defines the given word متسلسلة (Serialized) and  $w$  a target candidate تسلسل (Sequence), in case we don't find the n-gram  $w'_{3,4}$  of  $w'$  in the proper location the algorithm will shift the search to the right side in specific locations, so the n-gram  $w'_{3,4}$  will be compared first with the n-grams  $w_{3,4}$ , then  $w_{2,3}$  or  $w_{1,2}$  of the target candidate  $w$ , in case  $w$  greater than  $w'$  then the search will shift to left side. This will help also in case of misspelled words. Figure 3 show the similarity measure between the Arabic word التحالفات (the Alliances) and الفاتح (the Conqueror). Using the pure n-gram model, the similarity coefficient is quite high (85.72 %) although the two words do not belong to the same meaning class. This results from not taking into account the order of the n-gram on the target word. Figure 3 (right) shows the same example using the revised n-gram model. The similarity coefficient is quite low (28.57 %), since the order of n-gram was taken into account.

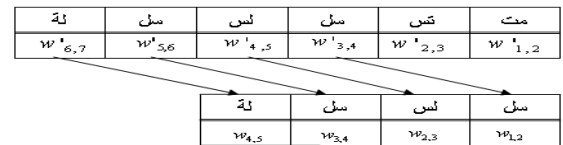


Figure 2. Words with different word lengths that belong to same meaning class

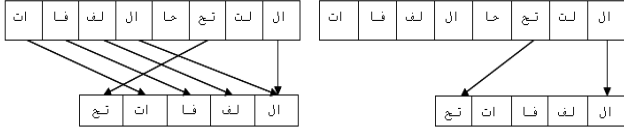


Figure 3. Pure bigram (left) and revised bigram (right)

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):

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

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

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

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 at a window of size  $m$  around the same position in word  $v$ .  $N$  is computed similarly as in Eq. (1).

## 4. Evaluation

In our experiments we compared our approach with the pure  $n$ -gram approach for bigrams and trigrams. The reason for not taking a larger value for  $n$  is the problem of eliminating short words. Previous Arabic studies demonstrate that the character  $n$ -gram with  $n=3$  or  $n=4$  are well suited for Arabic document retrieval. Thus, words with length less than 3 or 4 will not be retrieved, since for these no  $n$ -grams can be constructed. For example, when trying to retrieve the query **يقر** (Acknowledges) using trigrams, the relevant result **قر** (Acknowledged) will be eliminated because no  $n$ -grams can be constructed for it as it is less than 3 characters long. The targets words must be at least one character longer than the size of  $n$  in order to have the chance to be retrieved. For this reason, we used  $n=2$  in the proposed approach to enable retrieval of short words, as well as other words lengths. Furthermore, we used the revised  $n$ -gram model to avoid ambiguity as described above in Sect. 3.1.

### 4.1 Data selection

To collect test data for our evaluations, we crawled the web for articles published on one popular Arabic news Web site ("CNN-Arabic"<sup>1</sup>) in the period from January 2002 until March 2007 (for an example see Fig. 4). We thus obtained 5,792 Arabic documents, all of which are abstracts of articles on news, sport, art, economy and Information Science (size ~60MB). More than 1,400,000 Arabic words were extracted with 101,210 unique words. These articles are supposed to be correctly written and have both a large and rich vocabulary and therefore offer more

investigation points in terms of the number of word variations. The articles were carefully checked and cleaned.

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.



Figure 4. Example of an Arabic Document

### 4.2 Comparison of revised and pure $n$ -gram approaches

In a first experiment we calculated the average precision for each conflation approaches. Table 2 compares the result of the revised bigram and trigram approach with the result of the pure bigram and trigram models. As shown in the Table 2 the result are quite close. The reason for this is that only 6.5 % out of 500 queries words had a length of less than 3 characters, which is the length that affects the ambiguity. The revised bigram and trigram achieved a better improvement over the pure bigram and trigram due to the reduction of the ambiguity.

Table 2. Average precision for all approaches

Techniques	Precision
Revised bigram	92.28 %
Pure bigram	86.22 %
Revised trigram	98.74 %
Pure trigram	96.62 %

In a second experiment we calculated the average precision for the pure trigram and the revised bigram for the similarity thresholds of 60, 65, 70, 75, 80, 85, 90 and 95%. Table 3a and 3b show the comparison of retrieved, relevant, irrelevant and average precision between the revised bigram and pure trigram approaches. The revised bigram achieved clearly improvement over the pure trigram. The reason for that is that the revised bigram takes into account all words lengths which will increase the retrieved index terms size, on the other hand the it take into account the order of the  $n$ -gram which will decrease the pure  $n$ -gram ambiguity results. This will result in decreasing irrelevant terms retrieved. The trigram achieved better results in terms of the ratio of relevant index terms to the index terms retrieved. The revised bigram achieved better results in terms of how many relevant index terms were retrieved compared to the total number of index terms retrieved (relevant and irrelevant). For example, when selecting a threshold of 60 %, the revised bigram retrieved 5472 index terms relevant and 520 irrelevant, while the pure trigram retrieved 4253 index terms relevant and 189 irrelevant. The pure trigram retrieved less irrelevant index terms at the expense of the total number of relevant index terms retrieved while the revised bigram retrieved less irrelevant index terms compared to the total number of relevant index terms retrieved. It is important to notice, that when interpreting Figure 5c, one need to consider the big difference between the relevant index terms retrieved from each

<sup>1</sup> <http://arabic.cnn.com/>

method for different thresholds. As it is shown in Table 3a and 3b the performance of the revised n-gram approach is better than that of the pure n-gram in terms of the total number of relevant index terms retrieved. Table 4a and 4b provide a typical example where revised bigram model retrieved 33 relevant index terms while the pure trigram model retrieved 25 relevant index terms. In the second example, Table 4c and 4d show that the revised bigram model retrieved 18 index terms and all were relevant while the pure trigram retrieved only 8 relevant index terms. Figure 5a illustrates that although with a threshold of 85% both approaches have maximum precision, the revised bigram performs better than the pure trigram in terms of the number of relevant index terms retrieved.

**Table 3a. Average precision of pure trigram model for different thresholds on 500 words queries**

Threshold	Pure trigram			
	Ret.	Relev.	Irrelev.	Precision
60	4442	4253	189	0.957
65	3086	2969	117	0.962
70	2075	2045	30	0.985
75	1872	1843	29	0.984
80	1015	1007	8	0.992
85	549	549	0	1
90	549	549	0	1
95	549	549	0	1
Average Precision				<b>0.985</b>

**Table 3b. Average precision of revised bigram model for different threshold on 500 words queries**

Threshold	Revised bigram			
	Ret.	Relev.	Irrelev.	Precision
60	5992	5472	520	0.913
65	4367	4196	171	0.961
70	2960	2882	78	0.973
75	2464	2393	71	0.971
80	1817	1803	14	0.992
85	694	694	0	1
90	518	518	0	1
95	518	518	0	1
Average Precision				<b>0.976</b>

**Table 4a. The result of the query “مساعد” (helper) using the revised bigram approach**

Revised bigram approach			
S/N	Word	Rel/Irr	Translation
1	مساعد	Rel	Helper
2	بمساعد	Rel	By helper
3	بمساعدة	Rel	By help
4	تساعد	Rel	She helps
5	ساعد	Rel	He helped
6	ساعده	Rel	He helped him
7	ساعت	Rel	She helped
8	يساعد	Rel	He helps

9	كمساعدة	Rel	As a help
10	ومساعد	Rel	And helper
11	ومساعده	Rel	And his helper
12	ومساعدة	Rel	And help
13	وساعد	Rel	And he helped
14	لمساعد	Rel	For helper
15	لمساعدة	Rel	For help
16	نساعد	Rel	We help
17	مساعدتي	Rel	My helper
18	مساعدين	Rel	Helpers
19	مساعديه	Rel	His helpers
20	مساعدو	Rel	Helpers
21	مساعدون	Rel	Helpers
22	مساعدوه	Rel	His helpers
23	مساعده	Rel	His helper
24	مساعدها	Rel	Her helper
25	مساعداً	Rel	A helper
26	مساعدًا	Rel	A helper
27	مساعدات	Rel	Helps
28	مساعدة	Rel	Help
29	مساعدتي	Rel	My help
30	مساعدته	Rel	His help
31	أساعد	Rel	I help
32	المساعد	Rel	The helper
33	مساعدون	Rel	Helpers
34	ومساع	Irr	-
35	بمساع	Irr	-
36	لمساع	Irr	-
37	مساعي	Irr	-

**Table 4b. The result of the query “مساعد” (helper) using the pure trigram approach**

Pure trigram approach			
S/N	Word	Rel/Irr	Translation
1	مساعد	Rel	Helper
2	بمساعد	Rel	By helper
3	بمساعدة	Rel	By help
4	ساعد	Rel	He helped
5	كمساعدة	Rel	As a help
6	ومساعد	Rel	And helper
7	ومساعده	Rel	And his helper
8	ومساعدة	Rel	And help
9	لمساعد	Rel	For helper
10	لمساعدة	Rel	For help
11	مساعدتي	Rel	My helper
12	مساعدين	Rel	Helpers
13	مساعديه	Rel	His helpers
14	مساعدو	Rel	Helpers
15	مساعدون	Rel	Helpers
16	مساعدوه	Rel	His helpers
17	مساعده	Rel	His helper

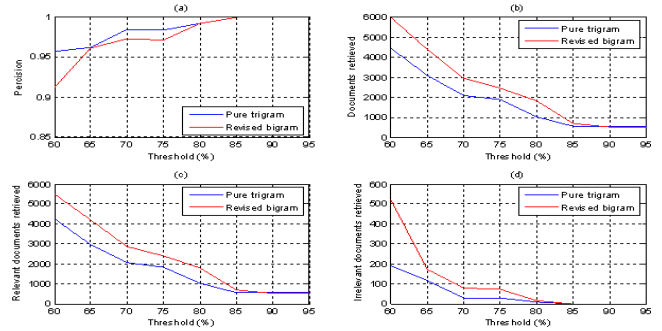
18	مساعدتها	Rel	Her helper
19	مساعدًا	Rel	A helper
20	مساعدًا	Rel	A helper
21	مساعدات	Rel	Helps
22	مساعدة	Rel	Help
23	مساعدتي	Rel	My help
24	مساعدته	Rel	His help
25	المساعد	Rel	The helper
26	مساع	Irr	-

**Table 4c. The result of the query “السياسة” (The politics) using the revised bigram approach**

Revised bigram approach			
S/N	Word	Rel/Irr	Translation
1	السياسة	Rel	The politics
2	السياسي	Rel	The Political (m)
3	السياسيين	Rel	The Politicians (m)
4	السياسيون	Rel	The Politicians (m)
5	السياسي	Rel	The Political (m)
6	السياسيات	Rel	The Politicians (f)
7	السياسية	Rel	The Political (f)
8	السياسات	Rel	The Policies
9	بالسياسية	Rel	By Political
10	بالسياسة	Rel	By politics
11	سياسة	Rel	politics
12	كالسياسة	Rel	As politics
13	وللسياسة	Rel	And for politics
14	والسياسي	Rel	And the Political (m)
15	والسياسية	Rel	And the Political (f)
16	والسياسة	Rel	And the politics
17	للسياسة	Rel	For politics
18	للسياسة	Rel	To politics

**Table 4d. The result of the query “السياسة” (The politics) using the revised pure trigram approach**

Pure trigram approach			
S/N	Word	Rel/Irr	Translation
1	السياسة	Rel	The politics
2	السياسي	Rel	The Political (m)
3	بالسياسة	Rel	By politics
4	سياسة	Rel	politics
5	كالسياسة	Rel	As politics
6	والسياسة	Rel	And the politics
7	للسياسة	Rel	For politics
8	للسياسة	Rel	To politics



**Figure 5. : a) - Average Precision. b) - Total index terms retrieved. c) - Relevant index terms retrieved. d) - Irrelevant index terms retrieved.**

In a third 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 Sect. 4.1. For all queries the number of relevant documents were obtained manually, by selecting all possible word variations. As shown in Tables 5a and 5b both approaches have very similar precisions, but the pure trigram approach missed many relevant index terms and therefore has a lower average recall than the revised bigram approach. The revised bigram approach gained up to 75% average recall while the pure trigram approach achieved 49%. Figure 6 illustrates that revised bigram gained a higher average recall than the pure trigram approach, since it took into account different words length and similarity enhancement. As shown in Tables 5a and 5b revised bigram approach gained a higher F-measure up to 76% compared to the pure trigram approach that gained 59%. These results show that the revised n-gram has gained an overall higher degree of retrieval performance than the pure n-gram approach.

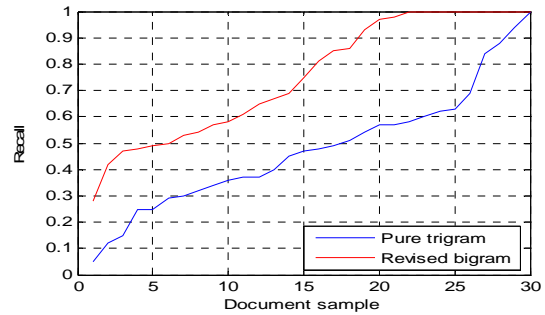
**Table 5a. Average Recall, Precision and F-measure for the pure trigram approach**

S/N	Pure trigram						
	Ret.	Rel.	Irr.	Miss. R.	Precision	Recall	F
1	7	6	1	7	0.85	0.47	0.61
2	6	6	0	11	1	0.36	0.53
3	17	17	0	13	1	0.57	0.73
4	1	1	0	2	1	0.34	0.51
5	29	28	1	0	0.96	1	0.98
6	10	9	1	11	0.90	0.45	0.60
7	22	22	0	3	1	0.88	0.94
8	13	13	0	23	1	0.37	0.54
9	7	7	0	22	1	0.25	0.40
10	6	6	0	14	1	0.30	0.46
11	1	1	0	19	1	0.05	0.10
12	6	5	1	11	0.83	0.32	0.23
13	3	3	0	23	1	0.12	0.46
14	11	11	0	8	1	0.58	0.73
15	14	14	0	24	1	0.37	0.54
16	1	1	0	6	1	0.15	0.26
17	14	13	1	6	0.92	0.69	0.79
18	18	17	1	19	0.94	0.48	0.64
19	16	16	0	14	1	0.54	0.70

20	28	28	0	2	1	0.94	0.97
21	10	10	0	6	1	0.63	0.77
22	10	10	0	30	1	0.25	0.40
23	11	11	0	17	1	0.40	0.57
24	20	20	0	13	1	0.60	0.75
25	12	12	0	8	1	0.49	0.66
26	12	12	0	30	1	0.29	0.45
27	2	2	0	2	1	0.51	0.68
28	38	38	0	19	1	0.57	0.73
29	16	16	0	10	1	0.62	0.77
30	5	5	0	1	1	0.84	0.91
	<b>366</b>	<b>360</b>	<b>6</b>	<b>374</b>	<b>0.98</b>	<b>0.49</b>	<b>0.59</b>

**Table 5b. Average Recall, Precision and F-measure for the revised bigram approach**

S/N	Pure trigram						
	Ret.	Rel.	Irr.	Miss. R.	Precision	Recall	F
1	9	7	2	6	0.77	0.54	0.63
2	7	7	0	10	1	0.42	0.60
3	28	26	2	2	0.92	0.93	0.92
4	3	3	0	0	1	1	1
5	29	28	1	0	0.96	1	0.98
6	13	12	1	6	0.92	0.67	0.78
7	25	24	1	0	0.96	1	0.98
8	36	35	1	1	0.97	0.98	0.97
9	15	14	1	15	0.93	0.49	0.64
10	10	10	0	10	1	0.50	0.67
11	7	5	2	13	0.71	0.28	0.40
12	18	16	2	0	0.88	1	0.94
13	12	12	0	14	1	0.47	0.64
14	29	19	10	0	0.65	1	0.79
15	38	38	0	0	1	1	1
16	4	4	0	3	1	0.58	0.73
17	20	13	7	6	0.65	0.69	0.67
18	18	17	1	19	0.94	0.48	0.64
19	21	17	3	13	0.80	0.57	0.67
20	29	27	2	1	0.93	0.97	0.95
21	16	16	0	0	1	1	1
22	27	26	1	14	0.96	0.65	0.78
23	17	17	0	11	1	0.61	0.76
24	28	28	0	5	1	0.85	0.92
25	27	23	4	0	1	1	1
26	22	22	0	20	1	0.53	0.70
27	3	3	0	1	1	0.75	0.86
28	49	49	0	8	1	0.86	0.92
29	30	29	1	7	0.96	0.81	0.88
30	6	6	0	0	1	1	1
	<b>596</b>	<b>553</b>	<b>42</b>	<b>185</b>	<b>0.93</b>	<b>0.75</b>	<b>0.76</b>



**Figure 6 Average Recall for Revised bigram and Pure trigram approaches (sorted by recall value)**

## 5. Conclusions

We presented a language independent conflation approach, i.e. the approach does not depend on any predefined rules or pre-linguistic information knowledge for the target language. We evaluated our approach on Arabic language which is one of most inflectional languages in the world. Since the previous Arabic studies demonstrated that n-grams of size 3 or 4 are the most suitable sizes for Arabic information retrieval, we focused on comparing our approach with trigram based models. The experimental results indicate, that the selection of the n-gram size affects the retrieval performance, i.e. the number of relevant and irrelevant documents retrieved. Using a big size of n lead to the fact that most of the documents retrieved are relevant but at the expense of missing many relevant documents, since the selection of a big n will eliminate short words to be considered. On the other hand, selecting a small value for n lead to the fact that many relevant documents are retrieved but at the same time many irrelevant documents are retrieved due to the ambiguity that is resulting of the small size of the n-grams. Therefore we proposed a revised approach to compare the similarity of words based on n-grams that take the order of n-grams into account. Based on the experimental results we could show that the revised bigram approach provided very good results compared to pure trigrams as well as n-grams with  $n > 3$ . Furthermore, we demonstrated that the enhancement of the n-gram model provided very good results in term of conflation for heavy inflection languages such as Arabic. Our algorithm was evaluated against 500 randomly selected queries. Unfortunately we had no benchmark results to compare our results with, but based on the quantitative and qualitative experimental results we could show that our algorithm achieved better results than pure n-gram approaches. Furthermore, our algorithm helps to achieve a higher degree of accuracy in the conflation task.

## 6. REFERENCES

- [1] Paice, C.D., 1990: "Another stemmer", *SIGIR Forum*, 24(3), 56-61 (Fall 1990).
- [2] Serhiy Kosinov. Evaluation of n-grams conflation approach in text-based information retrieval. *In 8th String Processing and Information Retrieval Symposium (SPIRE 2001)*, pages 136-142, 2001.
- [3] Ekmekcioglu, F. C., Lynch, M. F., and Willett, P. Stemming and n-gram matching for term conflation in Turkish texts. *In Information Research News*, 7 (1), pp. 2-6, 1996.

- [4] Al-Fedaghi Sabah S. and Fawaz Al-Anzi (1989) Anew algorithm to generate Arabic root-pattern forms. *Proceedings of the 11th National Computer Conference, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia.*, pp04-07.
- [5] Moukdad, H. (2004). Lost in Cyberspace: How do search engines handle Arabic queries? In Access to Information: Technologies, Skills, and Socio-Political Context. *Proceedings of the 32nd Annual Conference of the Canadian Association for Information Science, Winnipeg, June 3-5, 2004.*
- [6] Moukdad, H. and A. Large. (2001). Information retrieval from full-text Arabic databases: Can search engines designed for English do the job? *Libri 51 (2)*, 63-74.
- [7] Xu, Jinxi and Croft, W.B., "Corpus-Based Stemming using Co-occurrence of Word Variants" in *ACM TOIS, Jan. 1998, vol. 16, no. 1, pp. 61-81, Computer Science Technical Report TR96-67 (1996)*,.
- [8] Tai, S. Y., Ong, C. S., and Abdullah, N. A. On designing an automated Malaysian stemmer for the Malay language. (poster). In *Proceedings of the fifth international workshop on information retrieval with Asian languages, Hong Kong*, pp. 207-208, 2000.
- [9] Greengrass, M., Robertson, A. M., Robyn, S., and Willett, P. Processing morphological variants in searches of Latin text. *Information research news, 6 (4)*, pp. 2-5, 1996.
- [10] Berlian, V., Vega, S. N., and Bressan, S. Indexing the Indonesian web: Language identification and miscellaneous issues. *Presented at Tenth International World Wide Web Conference, Hong Kong, 2001.*
- [11] Carlberger, J., Dalianis, H., Hassel, M., and Knutsson, O. Improving precision in information retrieval for Swedish using stemming. In *Proceedings of NODALIDA '01 - 13th Nordic conference on computational linguistics. Uppsala, Sweden, 2001.*
- [12] Kraaij, W. and Pohlmann, R. Viewing stemming as recall enhancement. In *Proceedings of ACM SIGIR96*. pp. 40-48, 1996.
- [13] Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In Peters, C., Braschler, M., Gonzalo, J., Kluck, M., eds.: *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001. Volume 2406 of Lecture Notes in Computer Science.*, Springer (2002) 262–277
- [14] Moulinier, I., McCulloh, A., and Lund, E. West group at CLEF 2000: Non-English monolingual retrieval. In *Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop, C. Peters, Ed.*: Springer Verlag, pp. 176-187, 2001.
- [15] Popovic, M. and Willett, P. The effectiveness of stemming for natural-language access to Slovene textual data. *JASIS, 43 (5)*, pp. 384-390, 1992.
- [16] Khoja, S. and Garside, R. Stemming Arabic .Computing Department, Lancaster University, Lancaster, 1999 [www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps](http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps)
- [17] Larkey, L., Ballesteros, L. and Connell, M., "Light Stemming for Arabic IR," *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, A.Soudi, A. van den Bosch, and Neumann, G., *Editors. Kluwer/Springer's series on Text, Speech, and Language Technology* (2005).
- [18] Gelbukh, A., Alexandrov, M. and Han, S.Y.: Detecting Inflection Patterns in NL by Minimization of Morphological Model. In *CIARP 2004, LNCS 3287*, (2004) 432-438
- [19] T. Buckwalter. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0 [www ldc.upenn.edu/Catalog/CatologEntry.jsp?catalogId=LDC2002L49](http://www ldc.upenn.edu/Catalog/CatologEntry.jsp?catalogId=LDC2002L49).
- [20] M.F. Porter. An algorithm for suffix stripping. *Program, 14 (3)*: 130–137, 1980.
- [21] De Roeck, A. N. and Al-Fares, W. A morphologically sensitive clustering algorithm for identifying Arabic roots. In *Proceedings ACL-2000. Hong Kong, 2000.*
- [22] K. Darwish. An Arabic Morphological analyzer. <http://www.glue.umd.edu/~Kareem/research/>
- [23] G. Adamson and J. Boreham. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval, (10)*:253–260, 1974.
- [24] James Mayfield, Paul McNamee, Cash Costello, Christine Piatko, and Amit Banerjee, JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. In E. Voorhees and D. Harman (eds.), *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, Gaithersburg, Maryland, July 2002.
- [25] Darwish, K., & Oard, D. W. (2002). Term selection for searching printed Arabic. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR—2002)*, Tampere, Finland (pp. 261–268).
- [26] Suleiman H. Mustafa, 2004. Character contiguity in N-gram-based word matching: the case for Arabic text searching. *Information Processing and Management.41 (4)*, 819-827.
- [27] Laila Khreisat: Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study. *The 2006 International Conference on Data Mining Part of the 2006 World Congress in Computer Sciences DMIN 2006*: 78-82
- [28] Badam-Osor Khaltar; Atsushi Fujii; Tetsuya Ishikawa: Extracting loanwords from Mongolian corpora and producing a Japanese-Mongolian bilingual dictionary , *Annual Meeting of the ACL Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL Sydney, Australia* Pages: 657 - 664 Year of Publication: 2006
- [29] Farag Ahmed, Ernesto William De Luca und Andreas Nürnberger. MultiSpell: an N-Gram Based Language-Independent Spell Checker In: *Poster-Proceedings of Eighth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2007)*. (to appear).