Using N-Grams for Arabic Text Searching

Suleiman H. Mustafa and Qasem A. Al-Radaideh
Department of Computer Information Systems, Yarmouk University, Irbid, Jordan.
E-mail: {smustafa, qasemr}@yu.edu.jo

N-grams have been widely investigated for a number of text processing and retrieval applications. This article examines the performance of the digram and trigram term conflation techniques in the context of Arabic free text retrieval. It reports the results of using the N-gram approach for a corpus of thousands of distinct textual words drawn from a number of sources representing various disciplines. The results indicate that the digram method offers a better performance than trigram with respect to conflation precision and conflation recall ratios. In either case, the N-gram approach does not appear to provide an efficient conflation approach due to the peculiarities imposed by the Arabic infix structure that reduces the rate of correct N-gram matching.

Introduction

Word variation is one of the major challenges involved in free text searching. The most common types of variation that are encountered in textual databases are affixes, multiword concepts, spelling errors, alternative spellings, transliteration, and abbreviations. Several conflation techniques have been devised to handle these variations. As defined in the literature, conflation is the act of bringing together nonidentical textual words that are semantically related and reducing them to a controlled or single form for retrieval purposes.

Conflation techniques can be classified as being one of two major approaches: traditional and nontraditional (also known as algorithmic). Conflation has traditionally been performed by means of a comprehensive thesaurus. A thesaurus provides a precise and controlled vocabulary specifying the words and concepts of a given subject domain (together with their various conceptual and morphological relationships) that are to be used for indexing and searching.

Modern algorithmic conflation approaches, on the other hand, have emerged in response to the need for reducing the labor and cost involved in the manual generation of a carefully designed, reliable thesaurus and in response to the skepticism raised over the possibility of fully automating this process (Srinivasan, 1992). Algorithmic methods are quite complex and rely on different kinds of information ranging from linguistic rules of inflection and derivation to word pattern structure and its statistical decomposition (Kosinov, 2001). These approaches encompass three different categories of techniques: affix-removal stemming, string similarity measures, and successor variety counts. They represent documents and queries with free terms appearing in a given textual database.

There exists a vast literature on the principles, methodologies, and problems involved in the application of algorithmic conflation techniques to English textual databases. However, little attention has been devoted to conflation of textual data in other languages, especially with respect to the quantitative techniques. Most of the work reported in the literature concerning Arabic texts has focused on affix-removal stemming. The only reference to other conflation techniques has been reported by Mustafa and Al-Radaideh (2001), who applied the successor variety approach to Arabic text.

The primary goal of this paper is to investigate the performance of an N-gram conflation method within the context of Arabic textual retrieval systems. The method has been assumed to be language-independent and should work for all languages (Damashek, 1995; Huffman, 1995). It is therefore important to determine whether or not the results obtained thus far, with respect to English and other languages, are also applicable to Arabic text processing. The fact that Arabic is an agglutinative language with a complex affix structure involving prefixes, infixes, and suffixes presents a special case for testing this assumption.

String Similarity Measures

The string similarity measures approach to conflation is based on calculating a measure of similarity between a pair of words (i.e., an input query term and each of the distinct terms in the textual database). The work of Adamson and Boreham (1974) seems to have laid the foundation for this approach. Their underlying rationale was that the character structure of a word is so related to its semantic content as to

Received July 9, 2003; revised January 5, 2004; accepted January 5, 2004

© 2004 Wiley Periodicals, Inc. • Published online 30 March 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20051

make this a useful basis for automatic classification of words. The approach later became a topic of investigation for tasks related to information retrieval as early as in 1979 (Suen).

Depending on the application, affix-removal stemming methods share a number of drawbacks: They require a linguist or a polyglot for initial setup and subsequent tuning, they are vulnerable to variant spellings, misspellings, and random character errors, and they tend to be both language-dependent and domain-specific (Tan, Sung, Yu, & Xu, 2000). In comparison, it is believed that string similarity measures are more general in scope since they permit the conflation not only of morphological variants but also of transformation, spelling, and historical variants, inter alia (Ekmekcioglu, Lynch, Robertson, Sembok, & Willett, 1996).

The most commonly used string similarity measure is N-gram matching. An N-gram is an N character slice of a longer string. As defined in the literature, the term can include the notion of any co-occurring set of characters in a string (e.g., an N-gram made up of the first and third character of a word) (Cavnar & Trenkle, 1994). In this technique, the similarity between a pair of words is a function of the number of N-character substrings that they have in common. Based on ranking and using a threshold similarity of a given value, the N-gram technique groups words that contain identical character substrings of length N.

Different researchers have used different values for N. While some have reported the use of tetragrams (Damashek, 1995; Harding, Croft, & Weir, 1997), others have used digrams and trigrams. In either case, the results reported in the literature have indicated that the N-gram method provides better retrieval precision and recall performance than affix-removal stemming (Mayfield & McNamee, 1998). The general trend prevailing among researchers who have used the N-gram technique seems to indicate that digrams and trigrams provide acceptable substrings for N-gram matching.

It is, of course, possible to determine similarity using the occurrence of single characters as the attributes to be compared. But N-grams that are too short will tend to find similarities between words that are different. Words with the same root would be identified as matching when they did not, in fact, share a common root. Such erroneous conflation will decrease if larger N-grams are considered. However, N-grams that area too long will fail to capture similarity between different but similar words. This may mean that shorter common roots are missed or that spelling errors may lead to a large reduction in the number of common N-grams even with related terms (Freund & Willett, 1992; Gu & Berleant, 2000).

The N-gram method has been investigated in a number of ways. In some cases, its performance has been examined relative to that of affix-removal procedures (such as the Porter procedure) or successor variety stemming (Kosinov, 2001). In other cases, a combined approach has been used in which stemmed words (using an affix-removal stemming procedure) are input to the N-gram matching procedure (Ekmekcioglu et al., 1996a). An alternative way of combining N-grams and stemming is described by Croft and Xu (1995).

Since no prior linguistic knowledge about the text being analyzed is required by the N-gram method, it has been assumed to be language-independent. This characteristic has been confirmed by the results reported in the literature, with respect to languages other than English, such as Turkish, Malay, and Korean (Ekmekcioglu, Lynch, & Willett, 1996; Lee & Ahn, 1966). Along this line of thinking, some studies have been carried out to investigate the performance of the N-gram method in a multilingual setting (Cavnar & Trenkle, 1994; Ekmekcioglu et al., 1996a).

The N-gram method has been found useful in a wide variety of natural language-processing applications, including spelling error detection and correction (Harding et al., 1997; Peterson, 1980; Zamora, Pollock, & Zamora, 1981), text compression (Wisniewski, 1987), language identification (Damashek, 1995; Schmitt, 1990; Sibun & Reynar, 1996), text categorization (Cavnar & Trenkle, 1994; Huffman, 1995; Huffman & Damashek, 1994), text searching and retrieval (Cavnar, 1994), text retrieval from document images (Tan et al., 2000), and other information retrieval–related applications (Cavnar & Vayda, 1992, 1993).

Experimental Details

The N-Gram Procedure

In this paper, the approach being investigated was based on using consecutive digrams (with *n* = *2* letters) and trigrams (with *n* = *3* letters). Given a word like "الاستفسارات" *alestifsarat* (the queries), which is composed of eleven letters, the N-grams are generated as follows:

1. Digrams:

{"ال","لا","اس","ست","تف","فس","سا","ار","را","ات"}

2. Trigrams:

{"الا" ,"لاس","است","ستف","تفس","فسا","سار","ارا","رات"}

In general, a string of length *m*, has *m-1* such digrams and *m-2* trigrams.

It is worth noting here that some studies have appended leading and trailing blanks to the beginning and ending of the string in order to help with matching beginning-of-word and end-of-word characters. This would increase the number of N-grams for a given string. However, different studies have adopted different practices as to how these leading blanks would be applied. The following example shows how leading blanks for trigrams have been treated by two different studies:

B, BI, BIL, ILG, . . . , YAR, AR*, R

[Ekmekcioglu,et al., 1996b]

TE, TEX, EXT, XT, T** [Cavnar & Trenkle, 1994]

Our initial experimentation with the N-gram techniques did not support the idea of using leading or trailing blanks for N-gram-based Arabic string matching.

To test the applicability of the N-gram method to Arabic text searching, a sample of text consisting of six thousand (6,000) distinct textual words (i.e., not counting word frequencies) has been used in the experiment. The text came from a set of documents representing various disciplines that were extracted from the authors' own corpus. The procedure followed for generating the N-gram profile of the text involved the following steps:

1. Split the text into separate tokens consisting only of letters and insert them into a profile.
2. Sort the tokens and remove all duplicates, thus forming a new profile or a dictionary of distinct textual words.
3. Compute all the possible digrams and trigrams for each token and insert the two N-gram sets into the profile. Let these N-grams be denoted DG_w and TG_w consecutively (with w referring to a dictionary word).

To perform the string-matching process, a list of queries consisting of fifty distinct textual words was selected by systematic sampling. The procedure for matching these query words against the dictionary was as follows:

```
Repeat
1. Get the next query from the query profile.
2. Compute all the possible digrams and trigrams for the
   query. Let these N-grams be denoted $DG_q$ and $TG_q$
   consecutively (with q referring to a query word).
3. Go through the dictionary and compute the similarity
   value between $DG_q$ and $DG_w$ and between $TG_q$ and
   $TG_w$. The process is repeated for all items in the dictionary.
4. If the computed similarity value is greater than the similarity
   threshold specified (i.e., 0.6 in this experiment),
   insert the given token w into a list of suggested query
   variants (which may be referred to as an equivalence
   class, $E_q$). The list is sorted in descending order based on
   the similarity measures.
Until no more query words.
```

The string similarity measures for the two N-gram sets were calculated using Dice's Coefficient:

$$S = 2C_{wq}/(A_w + B_q) \qquad (1)$$

where:

S : the sought similarity value for a pair of words being compared *w* and *q*.

A_w : the number of unique N-grams in *w* (in this case, the dictionary word).

B_q : the number of unique N-grams in *q* (in this case, the query word).

C_{wq} : the total number of unique N-grams that are common to both words (the words "banana" and "bananas," for instance, have three common unique N-grams: "ba, an, na").

Given the Arabic word *alestifsarat* (the queries), mentioned above, and the word *estefsara* "استفسر" (queried), which consists of six letters and has the following digrams:

DG_q = {"سر","فس","تف","ست","اس"}

The similarity measure of the two words would be: *(2 * 4 /(10 + 5) = 0.533)*

Conflation Assessment

Words in an equivalence class E_q were assessed from two performance perspectives:

1. The degree to which retrieved words in E_q would be considered actual or relevant variants of the original query term *q*. Some of the words in E_q could be true variants while others could be false drops.
2. The degree to which the set of retrieved words in E_q covers all actual word variants that exist in the corpus data set. Some of the actual variants might be missing from E_q.

Intuitively, the two perspectives relate to the two commonly used measures in information retrieval: precision and recall. But since we were concerned with the number of words as a major indicator for string similarity assessment, rather than the number of documents, as used in precision and recall, we decided to use the two terms with some qualification. Hence, we refer to the first aspect by the term "conflation precision" (denoted CP), while we refer to the other aspect by the term "conflation recall" (denoted CR). The two measures would be defined formally as follows:

$$\mathrm{CP} = AV/TV \qquad (2)$$

$$\mathrm{CR} = AV/TAV \qquad (3)$$

where:

AV: Number of actual or relevant variants in E_q

TV : Total number of relevant and nonrelevant variants in E_q

TAV: Total number of actual or relevant variants in the corpus data set.

The first measure of effectiveness was simple to compute, since the values of the two parameters could be computed based on the equivalent class E_q. As to the other measure, some more work had to be done to arrive at the value of the dividend. For this purpose, we decided to save all the suggested variants with a similarity value less than the acceptable threshold. These variants were checked by

TABLE 1. Performance statistics for a set of query words using two N-gram methods (threshold = 0.6).

Word	Corpus TAV	Digram TV	Digram AV	Digram CR ratio	Digram CP ratio	Trigram TV	Trigram AV	Trigram CR ratio	Trigram CP ratio
أساسيات	33	13	12	0.36	0.92	5	3	0.09	0.60
أسعار	4	4	4	1.00	1.00	4	4	1.00	1.00
أسواق	4	4	3	0.75	0.75	3	2	0.50	0.67
أصول	7	7	6	0.86	0.86	6	5	0.71	0.83
الأمر	8	26	1	0.13	0.04	21	2	0.25	0.10
البلاغة	2	5	2	1.00	0.40	4	2	1.00	0.50
التابعة	7	7	4	0.57	0.57	3	3	0.43	1.00
التزام	7	8	5	0.71	0.63	3	3	0.43	1.00
التعادل	8	15	4	0.50	0.27	7	2	0.25	0.29
التفرقة	2	1	1	0.50	1.00	1	1	0.50	1.00

hand and the number of true variants, with a threshold < 0.60, was added to the actual number of variants in E_q, thus producing the total number of actual variants in the corpus data set.

For further checking, a set of stems for the query words were checked against the corpus data set in the dictionary using the N-gram procedure outlined above. Table 1 gives a set of ten query words (out of the total sample) and their performance values for the two N-gram methods.

The overall effectiveness *E* of the searches was measured by the mean values of the two measures *CP* and *CR* as follows (with *K* being the total number of queries included in the experiment):

$$E_{CP} = \sum(CP/K) \tag{4}$$

$$E_{CR} = \sum(CR/K) \tag{5}$$

The significance of the difference between the performance of the two types of N-grams (i.e., digrams and trigrams) was determined by means of the Sign test, a test of difference in location for two dependent groups. Chi-square was calculated as follows, with $df = 1$ (readers may refer to any textbook on statistics for details):

$$\chi^2 = (|f_{o+} - f_{e+}| - .5)^2/f_{e+} + (|f_{o-} - f_{e-}| - .5)^2/f_{e-} \tag{6}$$

where:

f_{o+}: obtained positive frequencies f_{e+}: expected positive frequencies

f_{o-}: obtained negative frequencies f_{e-}: expected negative frequencies

With $df = 1$, *Chi-square* (as determined by the χ^2 *Distribution*) must reach or exceed 3.84 to be significant at the 5% level, and 6.56 to be significant at the 1% level.

Experimental Results

An established fact in information retrieval is that recall and precision are inversely related. That is, when precision goes up, recall typically goes down and vice versa. The experimental results presented in Table 2, with respect to applying five different threshold values (i.e., 0.4, 0.5. 0.6, 0.7, and 0.8) to the N-gram method, indicate that 0.6 gave the best possible combination of conflation recall (CR) and conflation precision (CP) ratios. In Table 2, we find that 283 word variants out of 429 word variants retrieved (or, 5.66 out of 8.5 per query word on the average) were relevant. This is to be compared with the total number of relevant word variants in the corpus (i.e., 455 variants or an average of 9.1 per query word). The inverse relationship of the two ratios (CR and CP) is shown in Figure 1.

Using 0.60 as an acceptable threshold boundary, the mean performance values averaged over the number of query words (i.e., 50) for the two N-gram methods (digram

TABLE 2. The effect of using different threshold values on the performance of the digram method (total number of relevant variants in corpus = 455, i.e., an average of 9.1 variants per query word).

Threshold	Total retrieved	Avg. retrieved per case	Relevant retrieved	Avg. relevant per case	CP ratio	CR ratio
0.4	2,870	57.40	420	8.40	0.15	0.92
0.5	956	19.12	369	7.38	0.39	0.81
0.6	429	8.58	283	5.66	0.66	0.62
0.7	216	4.32	194	3.88	0.90	0.43
0.8	146	2.92	144	2.88	0.99	0.32

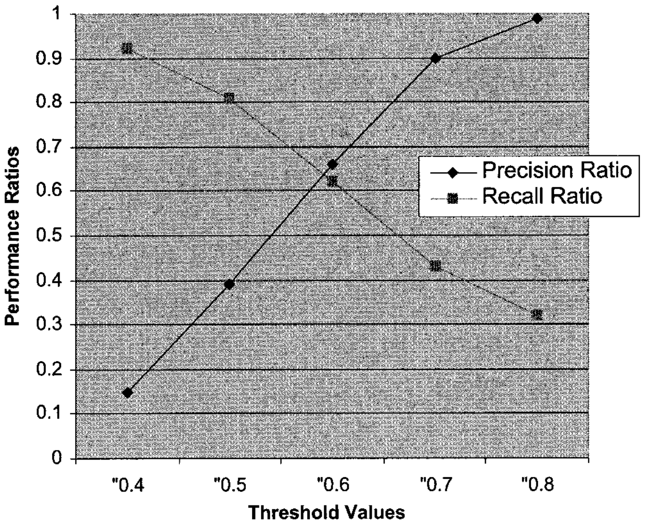

FIG. 1. The effect of using different threshold values on the performance of the digram method.

and trigram) are compared in Table 3. While the digram method offered an average performance of 5.66 relevant word variants per query word (out of 8.58 items retrieved), the trigram method offered an average of 4.0 relevant word variants per query word (out of 8.56 items retrieved).

This difference between the two N-gram methods is also reflected in the experimental results presented in Table 4. These results indicate that the digram method offers higher overall effectiveness values than the trigram method with respect to the two measures used: conflation recall (CR) ratio and conflation precision (CP) ratio. However, when these values were evaluated using the Sign test and Chi-square, the two methods were not found significantly different at the 0.5 level of significance.

Concluding Remarks

According to the experimental results, the N-gram conflation technique does not appear to provide an efficient approach to corpus-based Arabic word conflation. Our average rate of almost 30–35% of variants missing or in error in conflation precision and conflation recall (at 0.6 threshold boundary) raises questions of performance efficiency. Although the technique seems to be straightforward, the inherent lexical structure of the language under consideration still poses some problems on the applicability of the technique.

TABLE 3. Mean performance values averaged over the number of query words (i.e., 50) for two N-gram methods (threshold ≥ 0.6 and relevant average in corpus = 9.1).

	Digram		Trigram	
N-gram method	Total	Avg.	Total	Avg.
Variants retrieved	429	8.58	428	8.56
Relevant retrieved	283	5.66	200	4.0

TABLE 4. Conflation performance values for two N-gram methods, with threshold ≥ 0.6.

N-gram method	Conflation precision (E_{CP})	Conflation recall (E_{CR})
Digram	0.66	0.62
Trigram	0.47	0.44

These problems stem from the fact that most textual word variants involve a high rate of infix structure, which affects the computation of similarity measures. Two variants might have a similarity value far below the acceptable threshold factor, but in fact they are different only in terms of their infixes. In comparison, the Arabic infix lexical structure causes the performance of the N-gram technique to be lower than what has been reported in the literature with respect to other languages (such as English) and with respect to the algorithmic affix-removal stemming approach. An alternative approach for improving the results of the N-gram techniques could be to apply a two-step approach, in which the N-gram approach is combined with a stemming technique.

References

Adamson, G.W., & Boreham, J. (1974). 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.

Cavnar, W.B. (1994). N-gram-based text filtering. In D.K. Harman (Ed.), Proceedings of the Second Text Retrieval Conference (TREC-2) (pp. 171–179). Gaithersburg, MD: National Institute of Standards and Technology.

Cavnar, W.B., & Trenkle, J.M. (1994). N-gram-based text categorization. Proceedings of the Third Symposium on Document Analysis and Info (pp. 161–175). Las Vegas, NV: UNLV Publications/Reprographics.

Cavnar, W.B., & Vayda, A.J. (1992). Using superimposed coding of N-gram lists for efficient inexact matching. Proceedings of the Fifth USPS Advanced Technology Conference. Washington, DC.

Cavnar, W.B., & Vayda, A.J. (1993). N-gram-based matching for multi-field database access in postal applications. Proceedings of the 1993 Symposium on Document Analysis and Information Retrieval. Las Vegas, NV: University of Nevada.

Croft, W.B., & Xu, J. (1995). Corpus-specific stemming using word form co-occurrence. Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (pp. 147–159). Las Vegas, NV: University of Nevada.

Damashek, M. (1995). Gauging similarity with N-grams: Language-independent categorization of text. Science, 267, 843–848.

Ekmekcioglu, F.C., Lynch, M.F., Robertson, A.M., Sembok, A.M., & Willett, P. (1996a). Comparison of N-gram matching and stemming for term conflation in English, Malay, and Turkish texts. Text Technology, 6, 1–14.

Ekmekcioglu, F.C., Lynch, M.F., & Willett, P. (1996b). Stemming and N-gram matching for term conflation in Turkish texts. Information Research, 2(2). Retrieved June 2001, from http://informationr.net/ir/2-2/paper13.html

Freund, G.E., & Willett, P. (1992). Online identification of word variants and arbitrary truncation searching using a similarity measure. Information Technology: Research and Development, 1, 177–187.

Gu, Z., & Berleant, D. (2000, October). Hash table sizes for storing N-grams for text processing (Tech. Rep. No. 10-00a). Ames, IA: Iowa State University, Software Research Laboratory, Department of Electrical and Computer Engineering. Retrieved from http://citenseer.nj.nec-.com/347012.html

Harding, S.M., Croft, W.B., & Weir, C. (1997). Probabilistic retrieval of OCR degraded text using N-grams. Research and Advanced Technology for Digital Libraries. Proceedings of the First European Conference (ECDL), Pisa, Italy. Retrieved June 2001, from http://citeseer.nj.com/harding97probablistic.html

Huffman, S. (1995). Acquaintance: Language-independent document categorization by N-grams. Proceedings of the Fourth Text Retrieval Conference (TREC-4) (pp. 359–372). Gaithersburg, MD: National Institute of Standards and Technology.

Huffman, S., & Damashek, M. (1994). Acquaintance: A novel vector-space N-gram technique for document categorization. Proceedings of the Third Text Retrieval Conference (TREC-3) (pp. 305–310). Gaithersburg, MD: National Institute of Standards and Technology.

Kosinov, S. (2001, September). Evaluation of N-grams conflation approach in text-based information retrieval. InfoTech Oulu: International Workshop on Information Retrieval, Oulu, Finland. Retrieved June 2002, from www.syslab.ceu.hu/~serge/ir200/ir2001_kosinov_s.pdf

Lee, J.H., & Ahn, J.S. (1996). Using N-grams for Korean text retrieval. Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGR '96) (pp. 216–224), Zurich, Switzerland.

Mayfield, J., & McNamee, P. (1998). Indexing using both N-grams and words. Proceedings of the Seventh Text Retrieval Conference (TREC-7) (pp. 419–424). Gaithersburg, MD: National Institute of Standards and Technology.

Mustafa, S.H., & Al-Radaideh, Q.A. (2001, November). Arabic word stemming using letter successor and predecessor variety. Proceedings of the 2001 International Arab Conference on Information Technology (ACIT 2001) (pp. 216–222). Irbid, Jordan: Jordan University of Science and Technology.

Peterson, J.L. (1980). Computer programs for detecting and correcting spelling errors. Communications of the ACM, 23, 676–687.

Schmitt, J.C. (1990). Trigram-based method of language identification. U.S. Patent 5,062,143. Washington, DC: U.S. Trademark and Patent Office.

Sibun, P., & Reynar, J. Language identification: Examining the issues. Proceedings of the Symposium on Document Analysis and Information Retrieval (pp. 125–135), Las Vegas, NV.

Srinivasan, P. (1992). Thesaurus construction. In W.B. Frakes & R. Baeza-Yates (Eds.), Information retrieval: Data structures and algorithms (pp. 161–218). Englewood Cliffs, NJ: Prentice-Hall.

Suen, C.Y. (1979). N-gram statistics for natural language understanding and text processing. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 1(2), 164–172.

Tan, C.L., Sung, S.Y., Yu, Z., & Xu, Y. (2000). Text retrieval from document images based on N-gram algorithm. PRICAI 2000 Workshop on Text and Web Mining (pp. 1–12), Melbourne, Australia.

Wisniewski, J.N. (1987). Effective text compression with simultaneous digram and trigram encoding. Journal of Information Science: Principles and Practice, 13(3), 159–164.

Zamora, E.M., Pollock, J.J., & Zamora, A. (1981). The use of trigram analysis for spelling error detection. Information Processing and Management, 17(6), 305–316.