Evaluation of N-grams Conflation Approach in Text-based Information Retrieval

Serhiy Kosinov
University of Alberta
Computing Science Department
Edmonton, Alberta, Canada T6G 2E1
serhiy@cs.ualberta.ca

Abstract

This paper examines a conflation method based on the N-grams approach and evaluates its performance relative to the results achieved by other techniques such as Porter algorithm and successor variety stemming. In addition to that, an alternative way of enhancing the N-grams method, derived from the concept of inverse frequency weighing, is introduced and evaluated. The experimental results generated using standard collections ADI, CISI and Medlars show a consistent improvement over the traditional conflation methods, as well as demonstrate the viability of the introduced inverse frequency multiplier technique.

1. Introduction

As defined in the literature [10, 9], conflation is the process of matching non-identical words that refer to the same principle concept. In the context of information retrieval (IR), however, conflation has a more restricted meaning and usually refers to grouping together morphological variants of the same or related words. As such, conflation in textual IR helps overcome the problems of the strong dependence of the retrieval results on the exact wording of the user's information request, thus providing a way to account for the redundancy and richness of the natural language. From the point of view of IR system performance, various conflation techniques are most often regarded as a recall-enhancing device [15], since they expand the original query with related word forms, but they can sometimes improve precision as well by promoting relevant documents to high ranks. In addition to that, conflation brings another benefit of lowering system storage requirements by reducing the size of the indexing structures.

The primary goal of this paper is to study a conflation method based on the N-grams approach by evaluating its performance within textual information retrieval (IR) domain in comparison with other well-established techniques. The rest of this paper is structured as follows. The next section will briefly outline the related work previously done in the area of conflation methods. This will be followed by a description of the N-grams conflation approach, its proposed enhancements and details of implementation. Finally, the paper will be concluded with the sections on experimental results and future extensions of the presented method.

2. Selected related work

The majority of modern conflation approaches are quite complex and rely in their operation on many different kinds of information ranging from linguistic rules of inflection and derivation to the word pattern structure and its statistical decomposition. These conflation approaches can be subdivided into the following three major groups: successor variety, affix removal and N-grams methods.

As stated in [9, 13], the members of the first group (successor variety methods) are derived from a structural linguistic study of word and morpheme boundaries by applying its principles to sequences of letters within words rather than considering phonemes. The main hypothesis behind the successor variety is that the dependence between the letters is greater within certain word segments (such as morphological root) and less so between them. By identifying the boundaries of these word segments the method attempts to construct reasonable stems, hence the name of the technique known as successor variety stemming. As shown in [9], the successor variety conflation performs nearly as well as the most popular affix removal approaches, which are going to be described in the text that follows.

The next major group of the conflation approaches is the family of affix removal methods. As described in [13], these methods are based on such linguistic notions as roots and affixes, and, in particular in English, suffixes. In its operation, a typical affix removal method utilizes a number of morphological rules dealing with addition of suffixes to root forms, conversions between parts of speech, changes to plural form, deriving new words that are extensions of other

0-7695-1192-9/01 $10.00 © 2001 IEEE

Table 1. Pair-wise similarity calculation among three words: *photography, photographic, phonetic*

Words to compare (number of unique bigrams in brackets)	Common unique bigrams:		Similarity:	
Photography (9) and *Photographic (10)*	Ph ho ot to og gr ra ap	8	2*8/(9+10) =	0.84
Photography (9) and *Phonetic (7)*	Ph ho	2	2*2/(9+7) =	0.25
Photographic (10) and *Phonetic (7)*	Ph ho ic	3	2*3/(10+7) =	0.35

words, etc. Similarly, to the successor variety techniques, affix removal methods reduce a word to its morphological root or a stable stem, which is why they are called stemming algorithms.

Within the affix removal methods group, the most popular are the Porter[11] and Lovins[8] iterative rule-based stemming algorithms, which have been shown to improve recall indicators[5]. The results for precision are mixed, however, mostly because of the mistakes in reducing unrelated words to the same root, for example, as in the case with words "policy" and "police". To address these and other issues, various improvements were suggested. For instance, KStem algorithm [6], in addition to the morphological inference rules, uses machine-readable dictionaries to avoid problems such as the one mentioned above. Croft et al.[15] showed that corpus-based analysis of co-occurrence information also brings a stable, albeit not very large, performance gain. Nevertheless, in the area of the affix removal methods, the Porter algorithm remains a de-facto standard because of its simplicity, exceptionally high efficiency, and ability to achieve best results in the majority of applications.

Finally, the third major group of conflation approaches consists of the N-grams methods. The main idea of the N-grams approach, which groups together words that contain identical character sub-strings of length N called N-grams[1], is that the character structure of a word can be used to find semantically similar words and word variants. The approach assumes no prior linguistic knowledge about the text being processed, and thus is immune to spelling problems, which, as noted in [2], is an extremely helpful property for such applications as, for instance, optical character recognition (OCR). Furthermore, there is no language-specific information used in the N-grams approach either, which qualifies this method as a language-independent one. This latter characteristic of the N-grams approach was confirmed by the results of the studies of the technique in Turkish [3] and Korean[7] languages.

However, there has not been much published research on the application of the N-grams methods in IR and their performance relative to that of the above mentioned conflation approaches. This work attempts to address this issue by evaluating the precision and recall indicators of these techniques on a set of standard text collections, assessing alternative ways for possible performance improvement , and considering some efficiency aspects, which is described in detail in the following sections.

3. Methods and approaches used

3.1. Word similarity according to N-gram structure

As it was briefly mentioned in the previous section, the basic idea of the N-grams approach involves two simple steps. First, we subdivide words into N-grams - a set of overlapping sub-strings of length N. Second, we conflate similar words, i.e. the ones that have identical N-gram structure. In order to estimate the similarity of the words and make a decision as for which of them can be better candidates for conflation, Dice's similarity coefficient[4] for a pair of words is used. Its value is calculated according to the following formula:

$$S = \frac{2C}{A+B}$$

where S is the sought similarity value, A and B are the respective numbers of unique N-grams in word one and word two, and C is the total number of unique N-grams that are common for both words being compared. To illustrate the notion of N-grams similarity on a simple example with three words - photography, photographic, phonetic, - and N=2 (i.e. the case of bigrams), let us consider the data given in Table 1 [1]. The calculated similarity values listed in the last column of Table 1 show that the most similar pair of words among the three, and therefore the one that should be conflated, is, obviously, *photography* and *photographic.* In the proposed conflation method, these very same operations are carried out exactly as shown above, the only difference being that the system processes a much larger set of unique terms of a document collection.

3.2. Clustering techniques

Another indispensable routine used in the proposed N-grams conflation method is the clustering procedure. In order to see its importance for this work, it would be helpful to take a step back and consider what makes an N-grams approach different from the other conflation methods. So, if one views such techniques as successor variety or affix

[1] The example was adapted from [9]

removal algorithms as a group, it can be easily seen that all of them share an essential feature - they produce a stem for a word. Given this, all of the words that are reduced to the same stem form an "equivalence class"[2] and can be referred to by using the equivalence class representative, which is the common stem. On the other hand, when using N-grams approach, the situation is different. The similarity values that can be calculated from the set of unique terms of a corpus are only pair-wise estimates that provide no information on the size and structure of equivalence classes, or their representatives that can be used for matching documents and user queries. Therefore, in the N-grams conflation approach the formation of the equivalence classes must be carried out explicitly by using clustering techniques. This, however, still does not answer the question of deriving a class representative, but as it will be shown later, this issue may be resolved in a rather simple way once the structure of the equivalence classes is obtained.

As for the actual clustering procedure used in this work, the following aspects were considered. First and the foremost, when choosing an appropriate clustering method, one needs to ensure that a number of important adequacy requirements are met. According to [14], the clustering method must be stable under growth (i.e. a resulting cluster structure is unlikely to change drastically when more objects are added), robust (i.e. small errors in the description of objects lead to small changes in the clustering) and independent of the initial ordering of the objects. As proposed in [14], among the clustering algorithms for the IR tasks that satisfy the above criteria, the agglomerative hierarchical clustering (AHC) methods (such as single-link and complete-link algorithms) are considered the most suitable ones.

Furthermore, for the N-gram conflation method, hierarchical clustering may be more beneficial than others because of the advantage of ensuring that the performance in terms of recall could be at least as good as for unprocessed documents and queries. This observation can be explained by the fact that the algorithm starts out by putting every unique term in its own separate cluster, and, even in the worst case when no entities are successfully clustered, the resulting equivalence class structure would simply correspond to that of the original document and query collections.

Although recommended in several literature sources [14, 13], the single-link clustering algorithm according to the preliminary testing results proved to be inappropriate for the proposed N-gram conflation method due to its susceptibility to the chaining effect and tendency to produce overstretched and elongated clusters that grouped a lot of unrelated words together. On the other hand, the complete-link clustering algorithm was found to generate more compact and tightly bound clusters, which is why in this work it was given preference over the single-link clustering algorithm.

3.3. Finding a stem - is it really necessary?

It was mentioned earlier that in the majority of IR systems that incorporate stemming techniques, any given equivalence class is represented by the common to all of the members of the class stem. At the same time, all of the information that the members of a given equivalence class possess is ignored, because conventionally, the common stem replaces these class members in both queries and documents. In the case of the N-grams conflation method, the situation is quite the opposite. Having built the equivalence classes by using a clustering procedure outlined above, we do have explicit representation of all of the equivalence classes of words that were conflated, while the class representatives are not defined. A traditional approach [14] would suggest undertaking an extra stage of computations that would render a cluster centroid for each equivalence class (i.e. a stem), which in turn could be used for matching documents and queries later on. For this purpose, one of the standard methods to use is to search for a maximally linked or most similar entity in a cluster, and pick it as a cluster centroid.

However, analogously to the case when a stem replaces all of the terms that are members of a given equivalence class, by picking out a centroid in such a way we definitely lose information that all of the cluster members possess as a whole. Taking this argument even further, one may say that inherent coarseness of the decision scale adopted in the majority of the stemming-oriented IR systems may be detrimental to the retrieval outcome because of over-stemming and under-stemming errors. Put differently, the granularity of choice that a stemming algorithm faces, which is equal to one whole letter, may be considered too large to make a correct decision. To illustrate this problem by an example, we may say that for a sample equivalence class of three words - *divide, division, divisor,* - no stemmer program can produce a usable stem that would contain $\frac{2}{3}$ of letter "*s*" and $\frac{1}{3}$ of letter "*d*" in the last character of a representative stem (e.g. *divid/divis*) to reflect the structural information of the equivalence class in question; instead, it would have to take only one candidate ignoring the rest[3].

Taking these considerations into account, in the proposed N-grams conflation method all of the collected information about the created clusters, i.e. equivalence classes, was kept as is, and no additional computations for finding a suitable

[2] Of course, in the majority of the IR systems, these equivalence classes are never used explicitly as such; instead a typical system operates only on the common stems, i.e. equivalence class representatives.

[3] An obvious choice of selecting *"divi"* as a stem in the above example can be considered an over-stemming error as it may coincide with the similarly derived stem for such words as *divine, divination*.

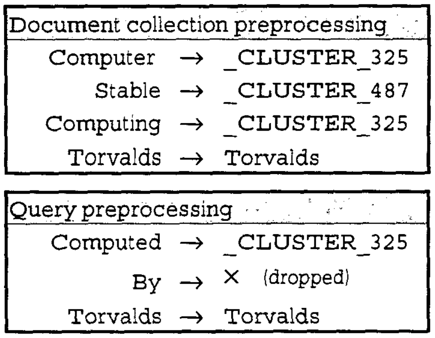
common stem for each of the clusters were carried out.

3.4. Query processing implementation with inverse N-gram frequency enhancement

From the technical point of view, the traditional scheme of query processing had to be changed slightly in order to accommodate the above premises. Namely, having the entire cluster set readily available, it was unnecessary to keep the original terms in the document collections, so they were simply replaced with cluster ID's they belonged to[4] As for the queries, the following technique was deployed. First, every term in a query was converted into a vector representation that reflected this term's N-gram structure, i.e. a vector with A^N components (where N is the order of N-grams used, and A - number of letters in the alphabet, 26 for English) corresponding to individual N-grams, each of which was equal to the number of times a given N-gram appeared in a term. Then, this vector was compared to the aggregated N-gram frequency information of each of the equivalence classes (clusters) represented by the very same type of vectors yielding a best-matching cluster ID (if any) according to the cosine similarity measure:

$$Sim_{Cos}(Cluster_i, term_j) = \frac{\sum_{k=1}^{A^N} a_{ik} \cdot a_{jk}}{\sqrt{\sum_{k=1}^{A^N} a_{ik}^2 \cdot \sum_{k=1}^{A^N} a_{jk}^2}}$$

where k is an N-gram index, a_{ik} is an aggregated N-gram frequency for k-th N-gram in a i-th cluster, a_{jk} - N-gram frequency for k-th N-gram in a j-th query term. If during these comparisons a significant similarity between the best-matching cluster and the query term was found, then this query term was replaced with the best-matching cluster ID. If the query term turned out to be a stop word, then it was dropped. Otherwise, the query term was left unchanged. This operation of documents/queries preprocessing is illustrated on Figure 1. Then, the process of matching queries and individual documents was carried out in a traditional way treating the equivalence class ID's (such as _CLUSTER_325) in documents and queries as ordinary terms subject to literal comparison, thus using the vector model once again, but this time in a conventional way by representing documents as vectors of terms.



Document collection preprocessing		
Computer	→	_CLUSTER_325
Stable	→	_CLUSTER_487
Computing	→	_CLUSTER_325
Torvalds	→	Torvalds

Query preprocessing		
Computed	→	_CLUSTER_325
By	→	× (dropped)
Torvalds	→	Torvalds

Figure 1. Document collection and query preprocessing example

In addition to the described above scheme, based on the standard vector model, an alternative way for improving the retrieval results was also introduced. The main idea of this enhancement technique is based upon the widely used concept of inverse document frequency (IDF) multipliers, which was applied in the domain of N-gram clusters. Thus, while N-gram information used in the cosine similarity measure formula above concerns only the occurrence of a given N-gram within a cluster, inverse N-gram frequency concerns the N-gram occurrence across the whole set of term clusters. The intuitive meaning of the inverse N-gram frequency is absolutely equivalent to that of the IDF: the N-grams that occur frequently in the whole cluster set are less valuable than those that appear not so often. Therefore, the importance of a given N-gram is assumed to be inversely proportional to the number of clusters that contain it. Practically, this method allowed to avoid incorrect association of certain terms based on the frequently occurring suffixes, such as *"-ing"*, *"-ate"* or *"-ation"*, etc. and let the morphological root of a given word weigh more. Formally, instead of pure aggregated N-gram frequency a_{ik} in the above Cosine similarity formula, its IF-weighted analog, w_{ik}, was used as shown below (considering the case of bigrams, i.e. N-gram order of N=2):

$$w_{ik} = a_{ik} \cdot \log_2 \frac{M}{m}$$

where w_{ik} - weight of k-th bigram in i-th cluster, a_{ik} - aggregated frequency of k-th bigram in i-th cluster, M - number of clusters in the equivalence class set, and m - the number of clusters where k-th bigram occurs at least once.

4. Experimental results

In the conducted experiments, a set of three standardized document collections (ADI, CISI and MED[12]) with predefined queries and known relevant answers prepared by human experts was used. Also, any additional preprocessing of the collections (such as special character filtering and stop word removal) was carried out in exactly the same way as documented in previous work (e.g. [9]), so as to avoid problems of dependence of the results on the stop-word list used, etc. Finally, the evaluation of the alternative conflation method was done using the same techniques, such as

[4] The unclustered entities, i.e. one-word clusters were left as is.

11-point and 3-point precision average over the range of defined recall levels. The majority of tools necessary for performing this experiment were supplied by the author of [9], and were used without modification.

The most extensive IR performance evaluation experiments of the proposed N-grams conflation method were carried out on the first dataset[5] , that is, ADI collection. As the results show, the N-grams conflation approach consistently outperformed all of the other methods by a positive margin. The below diagram on Figure 2 and the numerical data from Table 2 provide the details of these experiments.

The experimental results of dependence of the achieved performance on the order of N-grams, i.e. the value of N, are shown in Table 3. It is important to note, that when interpreting the figures given in Table 3, one needs to consider the possible influence of the selected AHC algorithm cutoff value. The diagram on Figure 3 clarifies this issue. As it is easily seen from the diagram on Figure 3, the performance of the N-gram conflation approach is strongly influenced by the choice of the AHC cutoff value, which determines the size and the number of clusters, or equivalence classes, that the method would generate by clustering the set of unique terms of a document collection. Although the solid black curve (N-gram method with inverse cluster frequency enhancement) appears to be above the light-gray dotted line (the performance of Porter stemmer) for all of the tested values, still the issue of the correct choice of the AHC cutoff is nevertheless very important for the purpose of achieving best performance. In the present implementation of the method, the cutoff value is derived from the observation of changes in the current cluster similarity scores. In particular, this value is set to the score after which the current cluster similarity drops most.

Another important message that Figure 3 conveys is that the introduced IDF-like enhancement in the proposed N-grams conflation method achieves its goal by improving the method's performance. Without this inverse cluster frequency multiplier enhancement (see the dashed gray curve on Figure 3), however, the performance of the method seems to be only slightly better than that of Porter stemming algorithm for some AHC cutoff values.

The summary of results for all of the three text collections is given in Table 4.

5. Conclusion

According to the obtained experimental results, the presented N-grams conflation method appears to be a plausible technique for improving IR performance. This work showed it to be able to overcome the choice granularity problem of the majority of stemming approaches thus avoiding information loss, enhance the achieved results with the inverse cluster frequency multipliers, and demonstrate better performance in comparison to other conflation methods such as Porter algorithm and successor variety stemming. Although the obtained performance gain was not very large and a deeper research in the crucially important for the method clustering procedures is necessary, the fact that N-grams approach is a language-independent technique makes the achieved results even more interesting and opens new prospects for research in the applications of the method for a number of languages other than English.

References

[1] 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.

[2] A. Brakensiek, D. Willett, and G. Rigoll. Improved degraded document recognition with hybrid modeling techniques and character n-grams. *In Proceedings of the 15th Intl. Conference on Pattern Recognition (ICPR)*, 4:438–441, September 2000.

[3] F. Ekmekcioglu, M. Lynch, and P. Willett. Stemming and n-gram matching for term conflation in turkish texts. *Information Research*, 2(2), October 1996.

[4] W. Frakes. *Stemming algorithms*. Prentice Hall, 1992.

[5] W. Kraaij. Viewing stemming as recall enhancement. *In Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 40–48, 1996.

[6] R. Krovetz. Viewing morphology as an inference process. *In Proceedings of the ACM SIGIR '93*, pages 191–203, 1993.

[7] J. Lee and J. Ahn. Using n-grams for korean text retrieval. *In Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland*, pages 216–244, 1996.

[8] J. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, (11):22–31, 1968.

[9] N. Maloy. Successor variety stemming: variations on a theme. 2000. project report (unpublished).

[10] C. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.

[11] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[12] URL. Adi, cisi, med test collections. *ftp://ftp.cs.cornell.edu/pub/smart/*.

[13] URL:. Course cs5604 lecture notes. *Virginia Polytechnic Institute and State University*. ei.cs.vt.edu/ cs5604/cs5604cnIN/IN-ConfMethods.html.

[14] C. J. vanRijsbergen. *Information Retrieval*. 2nd ed., Butterworths, 1979.

[15] J. Xu and W. B. Croft. Corpus-based stemming using cooccurence of word variants. *ACM Transactions on Information Systems*, 16(1):61–81, January 1998.

[5] The substantial memory and CPU time requirements of the presently used clustering algorithms did not permit to conduct the tests on other large-scale collections, beyond those three mentioned above.

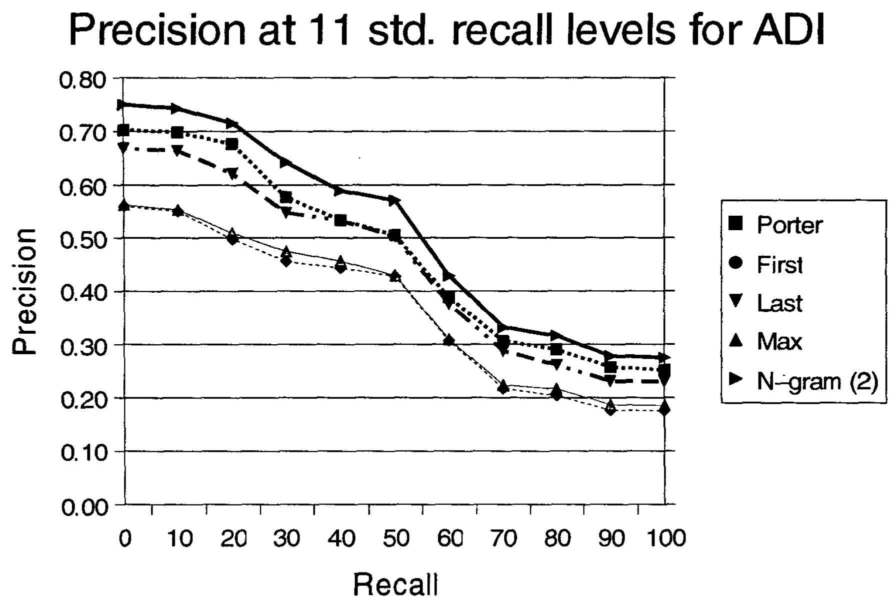


Figure 2. Precision at 11 standard recall levels on ADI collection for Porter stemming algorithm, First/Last/Max. Successor variety stemming and N-gram conflation approach (the case of bigrams, N=2)

Table 2. Precision at 11 standard recall levels on ADI collection for Porter stemming algorithm, First/Last/Maximum Peak Successor variety stemming and N-gram conflation approach (the case of bigrams, N=2)

Recall Level	Precision level				
	Porter algorithm	First peak	Last peak	Max peak	N-gram (2)
0	0.70287	0.55996	0.66784	0.56282	0.75039
10	0.69879	0.55043	0.66376	0.55330	0.74325
20	0.67658	0.49723	0.62102	0.50892	0.71578
30	0.57660	0.45596	0.54739	0.47434	0.64187
40	0.53337	0.44333	0.53267	0.45676	0.58913
50	0.50499	0.42747	0.50216	0.42928	0.57052
60	0.38771	0.30702	0.37550	0.30931	0.42883
70	0.30651	0.21706	0.28796	0.22378	0.33189
80	0.29046	0.20409	0.26114	0.21681	0.31623
90	0.25710	0.17685	0.23078	0.18729	0.27829
100	0.25151	0.17578	0.23012	0.18597	0.27497

Table 3. Dependence of N-gram conflation performance on the N-gram order (Porter stemmer indicators are given for comparison)

N-gram order	11-pt Precision Average	11-pt Precision Average
N=2 (bigrams)	0.51283	0.53418
N=3 (trigrams)	0.50407	0.53004
N=4 (4-grams)	0.50738	0.53132
Porter stemmer	0.47150	0.49068

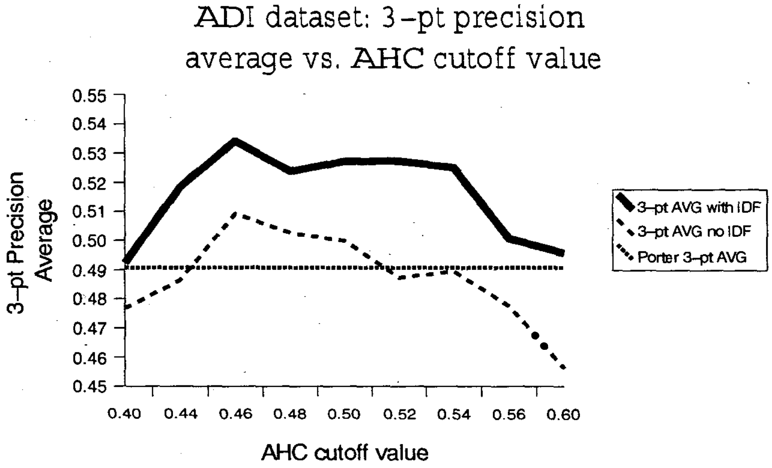


Figure 3. The influence of the agglomerative hierarchical clustering algorithm cutoff value on the performance of the N-grams conflation method (3-pt precision average).

Table 4. Summary of results on the three collections for N-gram conflation approach (N=2).

Collection	N-gram approach		Porter stemmer	
	11-pt precision avg.	3-pt precision avg.	11-pt precision avg.	3-pt precision avg.
ADI	0.51283	0.53418	0.47150	0.49068
CISI	0.16445	0.14715	0.15615	0.13534
ADI	0.56231	0.56925	0.54726	0.54906