Conflation Methods and Spelling Mistakes – A Sensitivity Analysis in Information Retrieval

Philipp Dopichaj, Theo Härder
Department of Computer Science, University of Kaiserslautern
P. O. Box 3049, D-67653 Kaiserslautern

Abstract

In some information retrieval scenarios, for example internal help desk systems, texts are entered into the document collection without proofreading. This can result in a relatively high number of spelling mistakes, which can skew the order of the documents retrieved for a query or even prevent the retrieval of relevant documents. We focus on addressing this problem at the conflation stage of the retrieval process and evaluate whether conflation based on n-grams, which is said to be insensitive to misspellings, leads to better retrieval quality than commonly used stemming algorithms. We do this by performing tests on artificially corrupted test collections and examine which characteristics of the queries and the relevant documents influence the relative retrieval quality achieved using the different conflation methods.

1 Introduction

The goal of textual information retrieval (IR) is to find from a set of texts information addressing a user's need.[1] Given a user *query*, a set of documents, the *document collection*, is searched, and a list of documents deemed relevant by the retrieval system is returned to the user.

In order to search the textual documents efficiently, they are usually transformed into a form more amenable to programmatic evaluation, the *document representatives.* There are several methods to extract the representatives from the original texts, the most popular ones – notably the *vector space model* – being based on dividing the text into words and deriving *index terms* from them. In the retrieval process, the terms used in the query are matched with the terms obtained from the documents.

The simplest approach uses the words themselves as the index terms. While this approach is sufficient in many contexts, it ignores the fact that words usually have several morphological variants, for example, *eating* versus *eaten.* For this reason, an additional step called *conflation* can be introduced to 'normalize' the words (see Section 2).

A problem arises if the texts are partly corrupted, for example through mistakes made by optical character recognition (OCR) systems or manual misspellings; even library personnel manually indexing books make spelling mistakes [Kla01], and it can be assumed that ordinary users posing queries to the system take less care and make more mistakes. These misspellings can prevent matching of terms, leading to severe problems, particularly if documents or queries are short.

There are several approaches to addressing this problem. In this paper, we will focus on a method used in the conflation phase of the indexing process; conflation based on n-grams is said to be insensitive to misspellings [Kos01], but so far there has been no thorough verification of this claim. For this reason, our goals are

[1] For a more thorough treatment of IR, see standard works such as [BYRN99].

- to determine to what extent conflation based on n-grams can be used to replace or enhance conventional stemming algorithms in the presence of misspellings,
- to examine how different levels of corruption affect the conflation methods, and
- what other factors (kinds of documents or queries, etc.) influence retrieval performance.

The next section will briefly introduce the conflation methods used by our evaluation, followed by the core section containing details about the experimental setup and an evaluation of the results. Finally, we summarize our findings and observations in Section 4.

2 Conflation Methods

The primary goal of conflation is to allow matching of different variants of the same word; in terms of standard IR quality measures, conflation improves *recall* (the quotient of the number of retrieved relevant documents and the total number of relevant documents). In addition to that, *precision* (quotient of the number of retrieved relevant and number of retrieved documents) can be positively affected, as several terms in the same documents can be conflated to the same index term, which can lead to a change in similarity to the query and thus the ranking. Furthermore, conflation can reduce the size of the document index significantly, because there are fewer distinct index terms that need to be stored.

Whether or not conflation can improve recall and precision depends on the characteristics of the documents and queries. In general, performance increases most if documents and queries are short. On the other hand, many conflation methods – in particular stemming algorithms like Lovins and Porter – have problems with misspelled words [Kro93], and the effect is greatest for short documents.

An example of retrieval on short texts is a library catalog: Usually, the full text of the books is not available in electronic form, so only short descriptions or even titles can be indexed. Another example is a help-desk system where experiences are stored in a knowledge base as they are made. In this scenario, the documents are not only short, but they are typically entered without any formal verification, so there may be a high number of misspellings. Furthermore, manually entered queries in most scenarios are typically only a few words long, so the corruption of just one word can significantly reduce the accuracy of the results.

2.1 Stemming Algorithms

The usual approach to conflation in IR is the use of a *stemming algorithm* that tries to find the *stem* of a word, that is the basic form from which inflected forms are derived. For example, the stem of both *eating* and *eaten* would be *eat*.

A common form of stemming is *affix removal* based on a list of affixes and rules. The best known implementations of this principle are the stemming algorithms of Lovins [Lov68] and Porter [Por80]. The difference between these two algorithms is that Lovins uses a longer suffix list whereas Porter relies on more complex rules.

2.2 n-Gram Word Conflation

Stemming algorithms only address the problem of morphological variants of words, ignoring the problem of misspellings. From a naïve point of view, these problems are similar: In both cases, the ‘original’ spelling of a word is slightly altered, yielding a new word form, so it is plausible that the methods used for spelling correction can also be used to handle inflection. One simple method for automatic spelling correction is to use so-called *n-grams* [AFW83], overlapping sequences of n letters extracted from the words.

Table 1: Properties of the test collections. Lengths are measured excluding stop words; short queries have a length of five words or less.

	Queries			Documents	
Collection	Avg. length	Number (short)	Redundancy	Avg. length	Number
ADI	8.2	35 (10)	8 %	35.1	82
CACM	13.3	52 (11)	20 %	35.2	3204
CISI	33.0	76 (10)	32 %	65.0	1460

Starting from this observation, a conflation method for IR has been developed: The words are clustered according to their similarity – which is calculated based on their n-gram structure –, and an identifier for the cluster replaces all occurrences of the words in the cluster. There are many conceivable variations: The value of n can be varied, and different clustering algorithms can be used; these algorithms, in turn, may have parameters that have to be tweaked.

We use the parameters chosen by Kosinov [Kos01], who evaluated such an approach and compared it to conventional stemming methods. He reached the conclusion that the new method leads to slightly better retrieval results, but he did not evaluate the effect of misspellings, even though he claimed that this approach is “immune to spelling problems”.

3 The Effect of Misspellings on the Quality of Retrieval

As we have seen, there are claims that conflation based on n-grams works well, irrespective of whether there are misspellings in the documents and queries. In this section, we present the framework used for verifying this claim, describe how we simulated the misspellings, and evaluate the test results.

3.1 The SMART Retrieval System and Test Collections

We used the SMART retrieval system and the included IR test collections, because it is freely available for research use.[2] Even though the system has not been updated since 1999, the vector space model it implements is still widely used. Additionally, the test collections that are distributed alongside it are well-suited to our task, because the documents and queries are rather short, so the effects of misspellings are significant and the time and space requirements for the tests are tolerable. The collections also include relevance judgements for all queries, which are used for evaluating the quality of the retrieval results using the recall and precision measures.

Table 1 lists relevant properties of the test collections; the redundancy of a query is the percentage of words in a query that occurs at least twice in the same query. In other words, if one word in the query is altered, the redundancy gives the probability that the original version of the altered word still occurs in the query.

3.2 Simulation of Misspellings

In order to evaluate how misspellings affect the quality of the retrieval results, we seed errors into the document and query texts and evaluate the change in the retrieval results depending on the conflation methods. The aim is to reveal what characteristics of the documents, queries, and misspellings are important for choosing an appropriate conflation method.

There are several reasons for misspelled words: The writer may believe a spelling to be correct even though it is wrong (for example, *comittee* instead of *committee*), or the word may simply be mistyped. The first problem cannot be simulated easily, because it requires long lists of correctly

[2] availabe at ftp://ftp.cs.cornell.edu/pub/smart

spelled words and their misspelled counterparts. Spelling errors, on the other hand, are more random; the great majority results in omitted, substituted, inserted or transposed (*gardneing*) letters [Mit96]. Thus, spelling errors can be simulated by applying one of these changes to a certain number of randomly chosen words from the documents.

Another parameter is where the mistakes occur, in the queries, the documents, or in both. Mistakes in documents can often be avoided by proofreading, or at least the negative effect of mistakes is low if the documents are long. Queries, on the other hand, are often formulated with little care, so we can assume that misspellings are much more common. This is particularly problematic because queries are typically much shorter than documents. For this reason and because the tests for corrupted documents are very time-consuming, our main focus is on the simulation of misspellings in the queries. The queries in our test collections are rather short, so it is not reasonable to insert a mistake into a fixed percentage of words, as this would mean that a large number of queries would be unchanged. Instead, we test a worst-case scenario, where *exactly* one misspelling is inserted into each query.

3.3 Evaluation

In order to determine the quality of the retrieval results, we measured the average precision at eleven fixed levels of recall for each combination of conflation method, test collection, and query.

Our first result is that spelling mistakes in the queries lead to a statistically significant decrease in retrieval quality: The p-values[3] obtained by comparing the accumulated results of all conflation methods with the paired Wilcoxon test are less than 0.01 for all test collections.

For the CISI collection, all conflation methods are roughly at the same level, and no statistically significant difference between them can be detected (all p-values are above 0.1). This level is almost retained when spelling mistakes are introduced, each method losing at most about 2.2 % of their original performance. The cause for this collection's insensitivity to spelling mistakes in the queries is most likely the high redundancy of the queries: In almost one third of the cases, a 'backup' copy of the altered word is present.

For the other collections ADI and CACM, the overall loss for the stemmers is higher at about 7–8 %, and n-gram conflation can reduce that to less than 3 %. As expected, the effects of corruption can be dramatic for short queries, where the stemmers lose about 13 % (ADI) and even 22 % (CACM). Under these circumstances, n-gram conflation shows its strength, losing only about 6 %. This difference is hard to quantify statistically, as the most sensitive test that is applicable, the paired Wilcoxon test, ignores pairs with a difference of zero. In our scenario, exactly these pairs are of importance, as they indicate that the conflation method managed to even out the differences due to misspellings. For example, with n-gram conflation in the ADI collection, only 6 out of 35 queries show any change at all, whereas with the Porter stemmer, 25 queries are affected.

In general, even though the n-gram conflation method is affected less by the introduction of spelling mistakes in the queries than the stemmers, its performance varies greatly from test collection to test collection: The results are excellent for the ADI collection, which is caused by conflating related words that are not morphological variants (for example, *biomedical* and *medical* [Kos01]). The flip side of this medal is that this can also lead to an effect similar to *overstemming*, that is, conflation of unrelated terms.

This problem occurs in particular using the CACM collection, which has more than 10 000 distinct words – compared to 1 300 in ADI – so that random collisions are common. One common example of this manifests in matching names with other, similar words so that queries seeking information about a certain person – which usually lead to good results – return seemingly random documents. Furthermore, long affixes like -*ability* can dominate the similarity of words with short roots, so that words like *recordability* and *liability* are considered equivalent.

[3] p-value: significance level; can be interpreted as the probability of the difference being caused by chance

Our rudimentary tests with corrupted documents (only on the ADI collection) indicate that the n-gram method's advantages vanish. The reason for this discrepancy (compared to corrupted queries) is that the clustering algorithm is too conservative, so that misspelled words frequently end up in clusters of their own, just as they do with the stemmers. The policy for matching query words to the clusters is less stringent, so the misspelled words are matched with the correct clusters in most cases.

Considering this, it seems reasonable to only use n-gram matching as a preprocessing step: The document corpus is indexed using a normal stemmer, but each query word is first replaced by the most similar unstemmed word from the document corpus (based on n-gram similarity), before the stemmer is applied to it. A brief evaluation indicated that this hybrid method exhibits roughly the same benefits with respect to misspellings as n-gram conflation.

4 Conclusions

The first thing to consider when deciding how to address spelling mistakes in an IR system is whether it is necessary at all. The differences in performance, while significant, are rather small for long queries and documents; only if queries are short, countermeasures are worthwhile.

The n-gram conflation method has shown to be less sensitive to the introduction of misspellings into the queries than the standard conflation methods, in particular, if the queries are short. It has to be considered, however, that the overall performance of that method varies significantly across the different test collections, that there are too many parameters that affect the retrieval quality, and that the clustering process required by this method is very time-consuming.

Using a hybrid approach appears to be more promising: Indexing can be achieved using a fast and well-tried stemmer, and n-gram matching for query words can reduce the negative effect of misspellings in queries; the advantage compared to a dictionary-based spell checker is that this method quickly adapts to the corpus-specific vocabulary without manual intervention.

References

[AFW83] Richard C. Angell, George E. Freund, and Peter Willett. Automatic spelling correction using a trigram similarity measure. *Information Processing & Management*, 19(4):255–261, 1983.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, 1999.

[Kla01] Henning Klauß. Tippfehler in Bibliothekskatalogen. *Bibliotheksdienst*, 35(7/8):868–876, 2001.

[Kos01] 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.

[Kro93] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM Press, 1993.

[Lov68] Julie B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[Mit96] Roger Mitton. Spellchecking by computer. *Journal of the Simplified Spelling Society*, 20(1):4–11, 1996.

[Por80] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.