Using contextual spelling correction to improve retrieval effectiveness in degraded text collections

Patrick Ruch
LITH-MEDIA Laboratory
Swiss Federal Institute of Technology
Lausanne - Switzerland

Abstract

The study presented relies on the design and evaluation of an improved IR system susceptible to cope with textual misspellings. After selecting an optimal weighting scheme for the engine, we evaluate the effect of misspellings on the retrieval effectiveness. Then, we compare the improvement brought to the engine by the adjunction of two different non-interactive spelling correction strategies: a classical one, based on a string-to-string edit distance calculus, and a contextual one, which adds linguistically-motivated features to the string distance module. The results for the latter suggest that average precision in degraded texts can be reduced to a few percents (4%).

1 Introduction

In *ad hoc* retrieval, the user enters a query describing the desired information, and the system returns a ranked list of documents. Although ad hoc information retrieval in textual repositories is probably the main application field of retrieval technologies, any knowledge extraction task (see for example, (Kageura et al., 2000) and (Klavans and Muresan, 2002)) share the same basic retrieval problem. In this paper, we argue that spelling errors are a major challenge for such systems, and evaluate the improvement brought by merging an IR engine with a contextual non-interactive spelling corrector, which uses linguistic representation levels.

Concerning spelling correction, common tools tolerate lower first guess accuracy by returning multiple guesses and allowing the user to interact with the system in order to make the final choice of the intended word. In contrast, some applications, like information retrieval, will require fully automatic correction for general purpose texts (Kukich, 1992). Thus, in the context of a "query by document" system, interactive spelling correction is forbidden for both queries and document collections.

In the study, we focus on IR models, and we do not use any model of common misspellings, the main reason is that we did not want to use knowledge about the error source (OCR, human...). Moreover, correction is restricted to the problem of non-existent words, and we do not attempt to solve corruption that results in valid, though unintended words (as in *piece* and *peace*, see for example Golding and Roth (1996)). If different studies (Mays et al. (1991), Brill and Moore (2000), Agirre et al. (1998)) showed that context sensitive correction will produce better results than isolated word error correction, then IR is probably the ideal application area for such tools.

The remainder of this paper is organized as follows. First, we present the major problems in the domains covered by the study. Second, we report on the tuning options of our IR system. Third, we evaluate the effect of misspellings on the IR system. Then, we measure the performance of the IR system by comparing results of the classical spelling corrector and the contextual one. The paper ends with some concluding remarks.

2 Problems

Assessing the effect of misspelled strings on an IR system implies to synthesize knowledge from at least three different origins: information retrieval, spelling correction, and information retrieval in corrupted collections.

2.1 Information retrieval

Any IR system defines four basic elements: a collection profile, a document and query representation item, a matching function, and a rank-

ing criteria.

Concerning the choice of the collection, there exist corpora with genuine spelling errors for the English language[1], but there are no IR collection (a set of queries associated with a list of relevant/non relevant documents) tailored for assessing misspellings effect on retrieval effectiveness. As it is easier to corrupt an IR collection than to develop an IR collection, the system was evaluated on a classical IR collection with artificially generated typing errors.

The CF collection[2], with 1239 documents and 100 queries, is a small IR collection. It has been chosen for the purpose of this study, because its query collection (with sometimes more than 30 tokens per query) could be regarded as a set of short documents, and therefore better simulate an IR engine accepting document extracts as queries. For the index content, we decided to use word stems[3], while the two last elements are dependent on the weighting features.

2.1.1 Weighting schemes

IR engines using a vector space approach are usually based on a variant of the *tf.idf* family (see Salton and Buckley (1988)). This approach states that the weight of a given term is related to the frequency of this term in a given document (i.e. the term frequency: tf), and inversely proportional to the frequency of this term throughout the document collection (inverse document frequency: idf). In table 1, we provide the most commonly used *tf.idf* features, following the SMART (Salton, 1971) representations.

A retrieval experiment can be characterized by a pair of triples *-ddd.qqq-* where the first triple corresponds to term weighting used for the document collection, and the second triple corresponds to the query term weight. Each triple refers to a term frequency, an inverse document frequency and a normalization function.

Depending on the collection, it is possible to calculate a posteriori the best weighting scheme. In these experiments, we limit our exploration

Term Frequency	
First Letter	$f(tf)$
n (natural)	tf
l (logarithmic)	$1 + log(tf)$
a (augmented)	$0.5 + 0.5 \times (tf/max(tf))$
Inverse Document Frequency	
Second Letter	$f(1/df)$
n(no)	1
t(full)	$log(N/df)$
Normalization	
Third Letter	$f(length)$
n(no)	1
c(cosine)	$\sqrt{\rho_1^2 + \rho_2^2 + ... + \rho_n^2}$

Table 1: Term Weights in the SMART System.

to a core parameter: *cosine normalization*. Cosine normalization is probably the more sensitive parameter of the tf-idf paradigm (see for example: Aizawa (2000)) and it plays an important role when applying IR systems to corrupted collections. Therefore we evaluate the IR system with and without cosine normalization. Let us note that cosine normalization is strictly applied at the level of the document collection: since normalization of query term weights just acts as a scaling factor for all the query-document similarities, and has no effect on the relative ranking of the documents, there was no need to vary the normalization factor for the query term weight.

2.2 Spelling correction

Spelling correction is processed by computing a string edit distance between a given token and the items of a lexical list (see Peterson (1980), for a survey of the probabilistic models of spelling). The main difficulty while applying spelling correction as a batch task is that the system may replace the misspelled token by a wrong candidate. Two systems are evaluated: the first system strictly relies on a dictionary of well-spelled tokens and select the top candidate based on a string edit distance calculus, while the second system uses lexical disambiguation tools in order to refine the ranking of the candidates.

2.3 OCR retrieval

Investigating retrieval and misspellings oblige us to refer to IR applied to optical character recognition (OCR) systems, whose most inter-

[1] http://titania.cobuild.collins.co.uk/boe_info.html

[2] The original CFC is available at http://www.sims.berkeley.edu/~hearst/irbook/cfc.html.

[3] Efficiency of the Porter's stemming is a matter of discussion in NLP (Krovetz, 1993)(Hull, 1996) applied to IR, but some preliminary tests showed that it performs well on the CF collection.

esting conclusions concern representation items of the IR engine and cosine normalization.

2.3.1 Representation items

The TREC 5 Confusion track, used a set of 49 known-item tasks to study the impact of data corruption (two corruption rates were applied: 5% and 20%) on retrieval system performance. A known-item search simulates a user seeking a unique particular, partially-remembered document in the collection. If there are obvious differences between known-item and ad hoc retrieval tasks, it is interesting to notice that retrieval methods that attempted a probabilistic reconstruction of the original text fared better than methods that simply accepted corrupted versions of the query text (Kantor and Voorhees, 2000): in particular engines using 4-grams as representation items did not perform very well.

2.3.2 Cosine normalization sensitivity

Misspellings not only create "garbage strings" (see Singhal et al. (1995) for an evaluation, and Mittendorf and Schauble (1996) for a more theoretical presentation), which will increase silence, but also corrupt the general document view formed by an information retrieval system, and therefore can substantially hinder the successful retrieval of relevant documents for user-queries. Indeed, most modern information retrieval system use sophisticated term weighting functions to assign importance to the individual words (or any other chosen items) of a document for document-content representation (Robertson and Jones (1976), Salton and McGill (1983)), and these term weight function, can be more or less dependent on the collection corruption.

Weighting functions use the occurrence statistics of words in the documents to assign importance to different words. As the occurrence statistics of words can change substantially due to OCR errors, weighting schemes are specially sensitive to degradation in the quality of the input text, and cosine normalization, which is commonly used in order to improve vector-space IR, must be manipulated carefully when applied to corrupted collection. Therefore, we have to decide whether cosine normalization can be used with our collection, i.e. if it would bring any improvement on the original CF collection as compared to a normalization-free weighting scheme.

3 IR engine tuning

In this section, we attempt to select an optimal weighting function for our collection. We concentrate on atn/atc parameters (see table 1: *augmented tf + idf +/- cosine normalization*), which are supposed to perform well on heterogeneous collection (Salton and Buckley, 1988).

3.1 Relevance scoring

In the CF collection, each query is provided with a ranked list of relevant documents. The ranking is provided by 4 experts along 3 relevance levels (0 = irrelevant, 1 = moderately relevant, 2 = very relevant), and results in a final relevance score, which ranges from 0 to 8. For the study, this fine-grained relevance score mapped to binary values (relevant or not) in order to be evaluated in a TREC-like style using the *treceval*[4] evaluation program.

3.2 Corruption model

We investigate the effect of misspellings both at the document and at the query level. Recent studies report on the high rate of misspellings in web user-query. Thus, Zeng and al. (2001) measure the rate of misspelled words in a large collection of queries issued from a medical web-based IR system and report a rate of 15%. While some other investigations, comparing newspaper articles and medical records (Ruch and Gaudinat, 2001) showed a minimal error rate of 3%, which goes up to 10% for the medical corpora. In the following experiments, queries were corrupted at 15%[5] and document at 3%. To perform the corruption, we define a corruption model, consistent with Damerau (1964)'s seminal researches. Damerau showed that 80% of misspellings can be generated from a correct spelling by a few simple rules: transposition of two adjacent letters (*heaptitis*), insertion of a letter (*heppatitis*), deletion of a letter (*hepattis*), replacement of a letter by another one (*hepatotis*).

[4] Available at: ftp://ftp.cs.cornell.edu/pub/smart/

[5] A 15% rate means that we randomly introduce a spelling errors every 6.66 words. This corruption rate is acceptable for short web queries but not for a query by document task. We must notice that degrading shorter queries -even with a more modest corruption rate- would have a much bigger effect on the retrieval evaluation.

```
# stems in the original collection: 6035
# stems in the misspelled collection: 11677
                                    (+93%)
# of relevant documents over all queries: 4801
```

Table 2: Index size and number of relevant document over all queries

	atn.ntn	atc.ntn
Rel_ret:	2249	2205
Interpolated Recall - Precision Averages		
at 0.00	0.8679	0.8290
at 0.10	0.6411	0.6219
at 0.20	0.5113	0.5033
at 0.30	0.3779	0.3470
at 0.40	0.2606	0.2369
at 0.50	0.1742	0.1680
at 0.60	0.0924	0.0894
at 0.70	0.0406	0.0332
at 0.80	0.0145	0.0140
at 0.90	0.0017	0.0013
at 1.00	0.0005	0.0000
Av. precision (non-interpolated) over all rel docs		
	0.2406 (100%)	0.2293 (95.3%)

Table 3: Comparison atn.ntn vs. atc.ntn

3.3 Characteristics of the IR system

Every experiment is conducted with Porter's conflation and using a list of English stopwords. Table 2 provides the index size calculated on each document collection: it is interesting to see how a low corruption rate can highly affects the main matching instrument of the IR engine: the index. For the corrupted collection this index is about twice (93%) as large as for the original one. This phenomenon is well documented in research conducted on OCR tools, and (Taghva et al., 1994) notice a huge increase (four fold) in the dictionary size for the degraded text collection.

3.3.1 Weighting schemes selection

In table 3, we compare retrieval effectiveness (interpolated recall at 11-pt and average precision, with N = 200) with and without cosine normalization, on the original collection (i.e. well-spelled queries and well-spelled documents). We observe that cosine normalization results in a moderate degradation at every point of recall. Therefore *atn.ntn* settings will now serve as a baseline for conducting the study, and we assume that IR effectiveness will be *qualitatively* affected in a similar fashion whatever

```
    Rel_ret:       2227
Interpolated Recall - Precision Averages:
    at 0.00       0.8479
    at 0.10       0.6270
    at 0.20       0.4927
    at 0.30       0.3632
    at 0.40       0.2452
    at 0.50       0.1742
    at 0.60       0.0932
    at 0.70       0.0370
    at 0.80       0.0154
    at 0.90       0.0018
    at 1.00       0.0005
Av. precision (non-interpolated) over all rel docs
                  0.2325 (96.6%)
```

Table 4: Results with misspelled documents

weighting scheme is applied[6]. It means that a precision of 24.06% represents the ideal result of the IR engine, therefore it represents the maximal precision (100%). Together with providing a baseline, this result allows us to restrict our studies to normalization-free weighting function.

4 Misspellings effects

Effects of misspelled words is measured along three modes: only documents are corrupted (table 4), only queries are corrupted (table 5), the whole collection is corrupted (table 6).

As expected the maximal degradation of the average precision is observed when both documents and queries are corrupted (table 6), with an average precision falling to 18%, i.e. 25% worst than for the original collection. Moreover, at any corruption level, the silence grows: from 2249 relevant documents retrieved originally (table 3: atn.ntn) down to 2227 (table 4), 2074 (table 5) and finally 2045 (table 6), i.e. misspellings make disappear at least 22 and at most 204 relevant documents.

Moreover, precision is almost uniformly affected, whatever interpolated recall point is considered. Even when only the document collection is corrupted (with a low corruption ratio: 3%), we observe a general average precision decrease of 4%. The degradation is even stronger (7%) when comparing average precision between tables 5 and 6 (we calculate the

[6] We do not pretend that these settings are optimal, and better ones, such as SMART Lnu-ltc or Okapi BM25, could be applied as well.

```
    Rel_ret:       2074
Interpolated Recall - Precision Averages:
    at 0.00       0.7537
    at 0.10       0.5364
    at 0.20       0.4127
    at 0.30       0.3090
    at 0.40       0.2069
    at 0.50       0.1386
    at 0.60       0.0691
    at 0.70       0.0290
    at 0.80       0.0077
    at 0.90       0.0032
    at 1.00       0.0005
Av. precision (non-interpolated) over all rel docs
                  0.1956 (81.3%)
```

Table 5: Results with misspelled queries

```
    Rel_ret:       2045
Interpolated Recall - Precision Averages:
    at 0.00       0.7276
    at 0.10       0.5050
    at 0.20       0.3888
    at 0.30       0.2901
    at 0.40       0.1824
    at 0.50       0.1278
    at 0.60       0.0664
    at 0.70       0.0246
    at 0.80       0.0070
    at 0.90       0.0019
    at 1.00       0.0005
Av. precision (non-interpolated) over all rel docs
                  0.1821 (75.7%)
```

Table 6: Results with misspelled queries and documents

ratio between each average precision: 19.56% and 18.21%).

The different deteriorations of retrieval effectiveness (3.4%, 18.7% and 24.3%), due to misspellings are somewhat comparable of the OCR deterioration reported in (Croft et al., 1994) and (Singhal et al., 1995), which respectively ranges from 1-10% to 9-14%, with a data corruption level of 5%, but restricted to the document collection. The difference in their results and the results from this study can be an artifact of the different collections that are being used in the two studies. Also, the text degradation schemes used in the two studies are substantially different. Overall, these results confirm the findings of Taghva and his colleagues that it is possible to use an automatic information retrieval system in conjunction with degraded text collections.

5 Results

For the final experiments on automatic spelling correction, we used a full-form 200 000-item dictionary. This dictionary was collected a priori by gathering all the English word frequency lists we could find and removing low frequency items. We did not perform any specific enrichment of the dictionary based on the CF collection. However the dictionary did incorporate medical vocabulary, as the SPECIALIST lexicon[7] was one of the collected lexical sources.

5.1 Common settings

We set up a confidence threshold[8] in order to avoid replacement of misspelled words by *bad* candidates: if the score of a candidate is below a certain edit distance, the replacement step is skipped. The basic idea is that we prefer to keep a misspelled word rather than replacing it by a wrong word. However, we could imagine a system, which would keep the original word in the query or in the document in addition to the replacement candidate. The entire IR task is very resilient to improper expansion of terms, therefore more than one candidate could be added.

5.2 Contextual spelling correction

We used the contextual spelling checker described in (Ruch et al., 2001). This system articulates three serialized modules. First, a string-to-string edit distance system, which provides a ranked list of candidates based on the string similarity to a full-form dictionary. Second, a part-of-speech tagger for English (Ruch et al., 2000), which considers the part-of-speech of the possible candidates and selects the best syntactic category based on the two previous and two following tokens. Third, a bigram word-language model, which finally ranks the list of candidates, based on word transition probabilities.

5.3 Measure

In table 7, column *context* measures the effect of applying the contextual spelling corrector on the corrupted collection. We observe the improvement in comparison with table 6, using

[7] http://www.nlm.nih.gov/research/umls/

[8] This threshold was determined experimentally as a ratio between the edit distance and the length of the token, the value was 0.3, allowing about 3 edit operations for a token of 10 characters

	context	str2str
Rel_ret:	2227	2198
Interpolated Recall - recision Averages:		
at 0.00	0.8501	0.7881
at 0.10	0.6217	0.5763
at 0.20	0.4952	0.4590
at 0.30	0.3649	0.3381
at 0.40	0.2474	0.2293
at 0.50	0.1645	0.1524
at 0.60	0.0886	0.0821
at 0.70	0.0384	0.0355
at 0.80	0.0145	0.0134
at 0.90	0.0017	0.0018
at 1.00	0.0005	0.0005
Av. precision (non-interpolated) over all rel docs		
	0.2319 (96.4%)	0.2232 (92.7%)

Table 7: Results for misspelled collection with contextual correction

the common baseline (calculated on the original collection) provided in table 3 (atn.ntn). Column *str2str* provides the same type of measure with the context-free spelling corrector. The average precision (96.4%) calculated with contextual correction is about 4% (3.6%) lower than the precision calculated on the original collection (table 3), and clearly outperforms the system using context-free correction, whose the precision is only 92.7%. Finally, the total number of relevant documents returned by the system using contextual spelling correction is the same (2227) as the one returned by a system applied on a document collection with 3% of misspelled words (table 4), and performs better than the context-free system (2198 relevant documents were retrieved).

6 Conclusion and future work

We showed that misspellings does affect retrieval effectiveness: already at very low corruption levels (3%), it affects the average precision of about 4% to 7%. In the worst case, when both queries and documents were corrupted, the average precision decreases of 25%, while with the help of the contextual spelling corrector, the retrieval degradation becomes of some percent only (about 4%). The average precision of an IR engine using a context-free corrector (7.3%) clearly underperforms the contextual one. In spite this relative success, it is difficult to compare these results with any previous studies, and the overall improvement brought by the contextual spelling checker over the non-contextual one is somehow modest regarding the resources it requires. Finally, considering that very often linguistically motivated IR techniques turned out to be no more effective than well-executed statistical approaches (Strzalkowski et al., 1998), IR in degraded text collection can be regarded as a promising research field for NLP.

From a more global point of view, merging a contextual spelling corrector with a retrieval engine requires a partial rethinking of both components. Until now, we showed how both systems can interact in order to improve retrieval, however we believe that a less additive and more synthetic interaction could bring substantial added-value. Therefore two research directions deserve to be mentioned.

6.1 Weighting and string normalization

The contextual spell-checker we used could be adapted to the collection the engine has to index. In this case, the correction could take advantage of the term frequency-inverse document frequency information, the idea is: when in the same document, a token A is found to be similar to a token B regarding the string distance, and it is observed that the frequency of A is high, while the frequency of B is low, then A is more likely to be the well-spelled instance of B. Such hypothesis could also be applied for static index pruning (Aizawa (1998), Carmel et al. (2001)), since misspellings does increase indexes' size. The same could be done at the stemming level, as misspelled instances cannot be mapped to the right stem.

6.2 Named-entity recognition

Another necessary step in order to build an IR system susceptible to cope with misspellings in unrestricted texts, concerns the handling of non-lexical items. Indeed, a collection of abstracts like the CF collection is rather poor in named-entities, while person's names and locations can be very frequent in other kinds of corpora, like newswires and literature excerpts.

References

E Agirre, K Gojenola, K Sarasola, and A Voutilainen. 1998. Towards a single proposal in spelling correction. *COLING-ACL Conference Proceedings.*

A Aizawa. 1998. Reducing the dimensions of attributes by selection and aggregation. *Lecture Notes in Artificial Intelligence 1532*, pages 417–418.

A Aizawa. 2000. The feature quantity: An information theoretic perspective of Tfidf-like measures. *Proc. of ACM SIGIR 2000*, pages 104–111.

E Brill and R Moore. 2000. An improved error model for noisy channel spelling correction. *ACL Conference Proceedings*, pages 286–293.

D Carmel, D Cohen, R Fagin, E Farchi, M Herscovici, Y Maarek, and A Soffer. 2001. Static index pruning for information retrieval systems. *Proc. of ACM-SIGIR*, pages 43–50.

W Croft, S Harding, K Taghva, and J Borsack. 1994. An evaluation of information retrieval accuracy with simulated OCR output. pages 115–26.

F Damerau. 1964. A technique for computer detection and correction of spelling errors. *ACM Comm.*, 7(3).

A Golding and D Roth. 1996. Applying winnow to context-sensitive spelling correction. In *Proc. of the Internation Conference on Machine Learning (ICML)*, pages 182–190.

D. Hull. 1996. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society of Information Science*, 47(1):70–84.

K Kageura, K Tsuji, and A Aizawa. 2000. Automatic thesaurus construction through multiple fitering. *Proc. of COLING 2000*, pages 397–403.

P Kantor and E Voorhees. 2000. The TREC-5 confusion track: Comparing retrieval methods for scanned text.

J Klavans and S Muresan. 2002. Evaluation of DEFINDER: A system to mine definitions from consumer-oriented medical text. *JCDL'01*.

R. Krovetz. 1993. Viewing morphology as an inference process. *Proc. of ACM-SIGIR*.

K Kukich. 1992. Techniques for automatically correcting words in text. *ACM Comp Survey, vol 24*, pages 377–439.

E Mays, F Damerau, and R Mercer. 1991. Context based spelling correction. *Information Processing Management*, 27(5):517–522.

E Mittendorf and P Schauble. 1996. Measuring the effects of data corruption on information retrieval. *SDAIR Proceedings*.

J Peterson. 1980. Computer programs for detecting and correcting spelling errors. *Commu. ACM*, 23(12).

S Robertson and K Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–46.

P Ruch and A Gaudinat. 2001. Comparing corpora and lexical disambiguation. *ACL Workshop on Comparing Corpora Proc.*

P Ruch, R Baud, P Bouillon, and G Robert. 2000. Minimal commitment and full lexical disambiguation: Balancing rules and hidden Markov models. *CoNLL-2000 (ACL-SIGNLL) Proceedings*, pages 111–115.

P Ruch, R Baud, and A Geissbuhler. 2001. Toward filling the gap between interactive and fully-automatic spelling correction using the linguistic context. In *Proceedings of the IEEE Workshop on Natural Language Processing and Knowledge Engineering (NLPKE)*.

G Salton and C Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–23.

G Salton and M McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill Book.

G Salton. 1971. *The SMART Retrieval System - Experiment in Automatic Document Retrieval*. Prentice Hall.

A Singhal, G Salton, and C Buckley. 1995. Length normalization in degraded text collections. Technical Report TR95-1507, NEC.

T Strzalkowski, G Stein, G Bowden Wise, J Perez Carballo, P Tapanainen, T Jarvinen, A Voutilainen, and J Karlgren. 1998. Natural language information retrieval: TREC-7 report. In *Text REtrieval Conference*, pages 164–173.

K Taghva, J Borsack, and A Condit. 1994. Results to applying probablistic IR to OCR text. *ACM-SIGIR*, pages 202–11.

Q. Zeng and al. 2001. Patient and clinician vocabulary: How different are they? *MedInfo'2001 Proceedings*.