Improving precision and recall using a Spellchecker in a search engine.

Mansour Sarr
NADA-KTH
Royal Institute of Technology
100 44 Stockholm, Sweden
Email: su98-mas@nada.kth.se

Abstract

Search engines are among the most popular tools for resource discovery on the Internet. Typically, users query a search engine by using natural language to extract documents that refer to the desired subject. Sometimes no information is found because they make spelling and typing mistakes while entering their queries. Earlier reports suggest that between 10-12 percent of all questions to a search engine are misspelled (Dalianis 2002; Stolpe 2003).

The issue is how much does the use of a spellchecker affect the performance of a search engine?

I will in this report present an evaluation of how much a query spellchecker improves precision and recall in information retrieval for Swedish texts. In order to make this feasible, I let under the course of two hours ten groups of students query the search engine SiteSeeker with and without the query spellchecker. Their task was to find the answers to some well-defined questions that were available in the corpus.

Evaluation results indicate that the spelling support improved both precision and recall with 4 respectively 11.5 percent.

1. Introduction

This report evaluates a spellchecker connected to the search engine SiteSeeker based on precision and recall. It is intended as an extension to the stemming experiment described in Carlberger et al (2001) and performed at the Department of Numerical Analysis and Computer Science (NADA) at the Royal Institute of Technology (KTH).

In their experiment they compared precision and recall with and without a stemmer. They concluded that stemming improved both precision and recall with 15 respectively 18 percent for Swedish texts having an average length of 181 words. This is described in more details later in this report.

The goal of my work was to set up an appropriate test experiment, evaluate it and then compare precision and recall with and without a query spellchecker.

1.1 Some background terminology

I would first like to introduce some terminology that will be used throughout this report.

- Query spellchecker

Spellcheckers generally process words as strings of characters. They identify misspellings by matching the string of characters of the misspelled word with strings of characters of words contained in the dictionary or index. If a match is not found, the query spellchecker flags

the word as a misspelling and generates a list of possible replacements. Matching the initial string of letters and applying morphological rules to possible replacements in the program dictionary identifies the word choices provided in a replacement word list for a given misspelling. Two factors influence the identification of the replacement word list: phonetic match and correct sequence of letters. The less severe the phonetic mismatch is to the target word the closer the query spellchecker will come to identifying the target word within the list of possible replacements because the string of characters of the misspelling will be similar to those of the target word.

❖ Spelling Errors

The word-error can belong to one of the two distinct categories, namely, *nonword error* and *real-word error*. Let a string of characters separated by spaces or punctuation marks be called a candidate string. A candidate string is a valid word if it has a meaning. Else, it is a nonword. By real word error we mean a valid but not the intended word in the sentence, thus making the sentence syntactically or semantically ill formed or incorrect. In both cases the problem is to detect the erroneous word and suggest correct alternatives.

In the case of typed text there are three kinds of nonword misspelling according to Kukich (1992): (1) typographic errors, (2) cognitive errors, and (3) phonetic errors.

In the case of typographic errors, it is assumed that the writer knows the correct spelling but accidentally presses the wrong key, presses two keys, presses the keys in the wrong order etc. (e.g., spell → speel). The source of cognitive errors (e.g., receive → recieve, minute → minite) is presumed to be a misconception or a lack of knowledge on the part of the writer. In the case of phonetic errors (e.g., naturally → nacherly, two → to) it is assumed that the writer substitutes a phonetically correct but orthographically incorrect sequence of letters for the intended word.

❖ Search engine

According to webopedia (2002) a search engine is a program that searches documents for specified keywords and returns a list of the documents where the keywords were found. Although search engine is really a general class of programs, the term is often used to specifically describe systems that enable users to search for documents on the World Wide Web and USENET newsgroups.

Typically, a search engine works by sending out a spider to fetch as many documents as possible. Another program, called an indexer, then reads these documents and creates an index based on the words contained in each document.

Each search engine uses a proprietary algorithm to create its indices such that, ideally, only meaningful results are returned for each query.

❖ Precision and Recall

These terms are metrics that traditionally define the "quality" of the retrieved documents set. In an ideal world, a perfect search will attain both high precision and high recall. In the real world of the Internet, a balance is struck, hopefully an optimum one.

Precision: *precision is defined as the number of relevant documents retrieved divided by the total number of documents retrieved.*

- For example, suppose there are 60 documents relevant to white roses in the collection. A query returns 40 documents, 30 of which are about white roses.
- 40 returned documents divided by 30 relevant ones = 75% precision

100% precision is an obtainable goal, since the system could be programmed to return just one completely relevant document (giving a 1/1 precision rate). However, returning just one document might not resolve the seeker's query. Therefore, a search system attempts to maximise both precision and recall simultaneously.

Recall: *recall is defined as the number of relevant documents retrieved divided by the total number of relevant documents in the index.*

- For example, suppose there are 60 documents relevant to white roses in the database. A query returns 48 documents, 30 of which are about white roses. The formula to calculate recall is would be:
- 60 relevant documents divided by 30 returned documents = 50% recall

Trying to accomplish a high recall rate on the Internet is difficult, due to the enormous volume of information, which must be searched.

In an ideal world, recall is 100%. However, since this is impossible to achieve since the system attempts to maximise both recall and precision simultaneously.

1.2 Motivation

So much information is now available on the Web that people must search through a plethora of accessible information in order to obtain meaningful information. Typically, users start a keyword-based Web search by using a search engine to extract documents that refer to the desired subject. Unfortunately, sometimes no documents are retrieved or many of the retrieved documents are irrelevant. The reason, according to Dalianis (2002) is because the word is not in the index or because the user misspelled the word, or because the user did not know the right inflection of the word as written in the index.

According to Kukich (1992), approximately 80% of all misspelled words contained a single instance of one of the four error types: insertion, deletion, substitution and transposition (switch to adjacent characters).

We also know that between 10-12 percent of all questions to a search engine are misspelled; these results are also supported in Stolpe (2003) and in a Google press release (2002).

For these observations and a claim made by Kann et al (1998) that up to a third of the search terms given to web based dictionaries are misspelled made us believe that attaching a spellchecker would assist the users a lot in terms of perhaps increasing either precision or recall or both.

A spellchecker is required to identify errors in queries where little or no contextual information is available and using some measure of similarity, recommend words that are most similar to each misspelled word.

This error checking would prevent wasted computational processing, prevent wasted users' time and hopefully make our system score high precision and recall.

Our query spellchecker performs a presence check on words against a stored lexicon, identifies spelling errors and recommends alternative spellings.

2. Previous research on SiteSeeker

Two major experiments have been conducted using the SiteSeeker search engine. Details on these can be found in Carlberger et al (2001) and Dalianis (2002). I will nevertheless give a quick overview on these experiments since our experiment is built on them.

2.1 The stemming experiment

This experiment was performed by Carlberger et al (2001). Their goal was to evaluate how much stemming improves precision in information retrieval for Swedish texts. They built an information retrieval tool with optional stemming and created a corpus of over 54,000 Swedish news articles.

Stemming as described in Carlberger et al (2000) is a technique to transform different inflections and derivations of the same word to one common "stem" (the least common denominator for the morphological variants). It can mean both prefix and suffix removal. Stemming can, for example, be used to ensure that the greatest number of relevant matches is included in search results. A word's stem is its most basic form: for example, the stem of a plural noun is the singular; the stem of a past-tense verb is the present tense.

The stemming algorithm for Swedish used about 150 stemming rules. The technique is about modifying the original word into an appropriate stem with a small set of suffix rules in a number of steps. The stemming

is done in one up to four steps and in each step no more than one rule from a set of rules is applied. This means that 0-4 rules are applied to each word passing through the stemmer. Each rule consists of a lexical pattern to match with the suffix of the word being stemmed and a set of modifiers, or commands. For further details see carlberger et al (2000) .

Over 54,000 news articles from the KTH News Corpus were selected (Hassel 2001). Among these, 100 texts were randomly selected and manually tagged a question and answer pair central to each text.

Three test participants conducted the experiment where a rotating questioning-answering evaluation schema was used. Each of the three participants answered 33, 33 and 34 questions respectively with and without stemming functionality.

After going through all of the 100 questions and finding answers to these, that is 33 questions each, the work was rotated and the participants became each others evaluators assessing how many of the found top ten answers were correct and how many were wrong.

Of the 100 questions, the test participants found 96 answers, 2 questions did not give any answers at all and 2 other questions gave unreadable files. Each of the asked queries had an average length of 2.7 words. The texts containing the answer had an average length of 181 words.

A 15 and 18 percent increase on precision respectively relative recall were found on the first 10 hits for stemming versus no stemming

2.2 The spelling support experiment

In this experiment, performed by Dalianis (2002), the website of the Swedish National Tax Board (RSV, Rikskatteverket) was used as a testing domain. This website is extensively used by the public to perform search on among others tax and income issues. The goal was to assess and evaluate the effectiveness of the query spellchecker from April to September 2001.

It should be mentioned that the spellchecking algorithm used stems from Stava and Granska (Domeij et al 1994, Carlberger & Kann 1999, 2000, Knutsson 2001) and makes use of the Edit-distance techniques described in (Kukich 1992).

During the five months period, the Swedish National Tax Board whose site contained roughly 6,000 documents used EuroSeek's search engine with built-in stemmer and dynamic query spellchecker. Over 1 million queries came in, out of which 101,446 (around 10 percent) were erroneous. Of the 100 most common spelling errors, the system gave 92 percent good suggestions and 40 percent among these contained split compound words, 22 percent were spelling errors and 30 percent were alternative spellings.

2.3 Other related work

Throughout the years, a number of experiments have been conducted on information retrieval and spellchecking. However no academic work detailing an evaluation of a spellchecker connected to a search engine that check the improvement on precision and recall could be found or made available. Instead many spellcheckers that have been built and evaluated are general purpose ones. These are suitable for any spellchecking application, even isolated word error correction like spellchecking user queries in a search engine.

A hybrid spellchecking methodology based upon phonetic matching and that is aimed towards being used in a search engine is to be found in Hodge & Austin (2001a). It aims to high recall accuracy at the expense of precision. Compare to several benchmark spellcheckers, the hybrid spellchecker did pretty well. It had the highest recall rate at 93.9% for a large 45,243-word lexicon and an increase recall to 98.5% for a smaller lexicon.

A method for detecting and correcting spelling errors in Swedish text was devised in Domeij et al (1994). It was then refined in Kann et al (1998) where a ranking of correction using word frequencies and editing distance was implemented. According to them, their spelling correction can be used in information retrieval where the users would for example be offered interactive spelling correction of misspelled search terms. This, they claim, would improve search results both as regards precision and recall.

Our spellchecker has been, in fact, put to the test twice. The first time it was used was in a Swedish-English web dictionary, which contains 28,500 Swedish words. About 20,000 queries were posed to the web dictionary every day. Of these 20% were misspelled. For 33% of the misspellings a single key is at closest distance to the misspelling, so the query could be corrected automatically.
The second time it was used is described in section 2.2.

In Hodge & Austin (2001b) a phonetic spellchecker—Phonetex—which is intended to integrate with an existing typographical spellchecker is evaluated. It was compared against other phonetic and benchmark spellcheckers. Different lexicon sizes were used to investigate the effect of lexicon size on recall accuracy using the test set of 360 phonetic misspellings. It was shown that Phonetex has the highest recall, only failing to find eight words from 360. It maintains high recall across lexicon sizes for best match retrieval ranging from 0.96 for the large 45000-word dictionary to 0.98 for the smaller dictionaries.

3. Experiment

From a sub-corpus of over 70,000 news articles fetched from the KTH News Corpus (Hassel 2001), 100 texts were randomly selected. Each text is manually tagged as a question and answer pair as described in

Carlberger et al (2001). We use these texts in our experiment too. An example is given in figure 1.

In this section, the experimental setting that is used to evaluate the performance of our system, is presented. Section 3.1 describes our test participants. Section 3.2 shows how the question and answer form were built and section 3.3 describes the set of rules that the test participants had to follow during the experiment.

Question
<top>
<num> Number: 35
<desc> Description: (Natural Language question)
Vem är koncernchef på Telenor? (Who is CEO at Telenor?)
</top>
Answer
<top>
<num> Number: 35
<answer> Answer: Tormod Hermansen
<file> File: KTH
NewsCorpus/Aftonbladet/Ekonomi/0108238621340_EKO__00.html
<person> Person: Tormod Hermansen
<location> Location: Norden
<organization> Organization: Telenor
<time> Time: onsdagen
<keywords> Keywords: Telenor; koncernchef; teleföretag;
mobilmarknaden; uppköp
</top>

Figure 1. Questioning and answering tagging scheme

3.1 The test participants

To simulate actual people querying a search engine as it is in the Web, we asked the students taking a course in language engineering to be our test panel. Some of them have Swedish as their first language others do not. They are familiar with search engines and have good knowledge in among others natural language processing. Their assignment was to, simply stated, query the search engine in order to find the answers to some asked questions. We split them into ten groups; each group comprises two participants.

3.2 Building the questions and answers forms

Since our test panel was divided into ten groups of two, our original 100 questions mentioned at the beginning of this section were also divided into ten questions. Group one would handle the first ten questions, group two the second ten questions and so on. These questions were made available to them only when experiment began.

We then built two types of answering forms for all 100 questions: one for answers from search with spellchecking and another for answers from search without spellchecking. For spellchecking a form contained a couple of fields. The first is a name field where the test participants would write their names, a second one where they would write the query words used. A third one for eventual query words suggested by SiteSeeker that they used. A fourth one was for the number of hit returned by the search engine. Finally two fields where they would write down the found answer respectively the link that gave that answer. A non spellchecking form does not have the third field mentioned above.

These answering forms, the location of the search engine and some couple of instructions were made available to the test participants via a web page.

3.3 The experiment rules

The experiment had to be conducted in two hours. During the first hour the test participants in each group had to query the search engine (with spellchecker) and find the answers to their ten questions. Actually, one of the two in each group read the questions while his/her partner did the searching and filled the answers they found in the form. The participant who was querying the search engine could not see how words were spelled. This simulates an ordinary web search where a user, wanting to find information on a news event or a subject, just types in the term that comes to his mind.

Some sets of rules were made that the test participants had to follow while querying:

1. They were not allowed to do more than five trials on each question to find the answer.
2. They were not allowed to use longer queries than five words.
3. No background knowledge was allowed, that is only the words used in natural language as they were written in the questionnaire were allowed. So even if the test participant were very familiar with a certain event and probably knew the answer to a certain questions, the latter is only allowed to use the words that were read to him/her.
4. Boolean expressions and phrases searches were allowed but were rarely used.
5. Each question had to be read no more than five times.
6. The full URL of each found answer to a question had to be written down for use during the evaluation process.

After having finished this first part, we rotated the question forms to avoid training effects of running the same set of questions. That is we did not want the test participants to query the search engine with the same set of questions whose answers they already knew. Now under the second hour we switched the question forms. Group one's question forms were given to group two and vice versa, group three's questions forms to group four and vice versa and so on. This time each group had

to query the search engine without the spellchecker and using the non-spellchecking answering form described in section 3.2 . The role of the participant in each group was also inverted in the second part of the experiment. Those who, during the first hour, read the questions to their partner queried the search engine and vice versa.

When all groups finished the second part, we made sure that they filled the two answering forms as correct and legible as possible just by browsing through them.

These forms will be used to evaluate our search engine based on precision and recall.

4. Evaluation

In this Section we present the evaluation of the experiment and our findings. Section 4.1 details the evaluation metrics for the performance of our query spellchecker. Section 4.2 introduces the parameters used to judge the relevancy of the returned documents by the search engine for a query q. Our final results are reported in Section 4.3

4.1 Evaluation Metrics

Information retrieval systems are usually compared based on the "quality" of the retrieved document sets. This "quality" is traditionally quantified using two metrics, *recall* and *precision*. Each document in the collection at hand is judged to be either *relevant* or *non-relevant* for a query. *Precision* is calculated as the fraction of relevant documents among the documents retrieved, and *recall* measures the coverage of the system as the fraction of *all* relevant documents in the collection that the system retrieved. See Section 1.1.4 for a more elaborate definition.

To measure recall over a collection we need to mark every document in the collection as either relevant or non-relevant for each evaluation query. This, of course, is a daunting task for any large document collection like ours, and is essentially impossible for the web, which contains billions of documents.

Due to the difficulties of calculating recall, *relative recall* will be used instead. Relative recall can be define as the fraction of known relevant items retrieved by the system. The following example will help to clarify this term:

1. System A retrieves 4 items: D1, D2, D3 and D4.
2. System B retrieves 3 items: D2, D3 and D5.
3. Known unique relevant items D1, D2, D3, D4 and D5 are assumed to approximate the total number of relevant items (5 in this case).
4. Relative recall for A: 4 / 5 = 80%
5. Relative recall for B: 3 / 5 = 60%

In this report, the results found in each search method are pooled and assumed to approximate the total number of relevant documents, and the relative recall of individual search engines is calculated from this.

4.2 Evaluating retrieved documents and their Relevance Judgements

What is a "relevant" document, and who determines the relevance of retrieved documents? In reality the only person who can determine whether a document is relevant to an information need is the person who has the information need, in this case it is us.

The answering forms described in Section 3.3 that the students have handed in are carefully scrutinised since these will determine how much precision and recall have increased. In order to verify the correctness of the answers given in the answering forms, we used, for each question, the actual query words that each group wrote down and query the search engine again. That way we could beside the document that contained the answer, also check the other nine of the top ten documents (or all documents if fewer than 10) returned by SiteSeeker.

We set up an evaluation scheme that presents the result pages for a query one page at a time, and allow us to judge each page as “good”, “bad” or “ignore” for the query.

We used the following criteria to judge documents. A good page is one that contains an answer to the test question *on the page*. If a page is not relevant, or only contains links to the useful information, even if links are to other documents on the same site, the page is judge “bad”. If a page could not be viewed for any reason (e.g., server error, broken link), the result is “ignored”.

4.3 Evaluation Results

In this section we report the results of the experimental evaluation using the methodology described in the previous section.

We use an Excel file and input for each test question the top ten retrieved documents when the non spellchecking was used in one hand and likewise when the spellchecking was used on the other hand.

Making use of the relevance judgement described in Section 4.2, we award each “good” page or link with 1. A “bad” one is awarded with 0.

Now using these figures and the metrics described in Section 4.1, we calculated precision and relative recall in both cases. Consequently, we found an increase on precision and relative recall with 4 percent respectively 11.5 percent (see Table 1).

Table 1. No Spellchecking versus Spellchecking

Precision/Recall At 10 first	**No Spellchecking**	**Spellchecking**
Number of texts	79,000	79,000
Number of questions	100	100
Average precision **Increase in precision %**	0.485	0.505 **4%**
Average relative recall **Increase in relative recall %**	0.52	0.58 **11.5%**

The number of misspellings that SiteSeeker gave suggestion links for were 14.

Concentrating on the nature of these misspellings, we noticed that 6 were of typographic type, 3 of cognitive type and 4 of phonetic type. Of all these, 13 were good suggestion links, that is they led to the correct answer. In 12 cases, more than one suggestion link was given. Interestingly only twice was the correct answer not the first suggestion. In both these two cases was the correct answer the second suggestion. This tells us that our system is good at ranking corrections; giving the most probable correction the first alternative in 10 cases out of 12. Only in 3 cases did a search with the spellchecker fail to return hit pages.

5. Conclusions and future improvements

We found in our experiment that the spellchecker connected to our search engine SiteSeeker does improve both precision and relative recall with 4 respectively 11.5 percent; and makes the interface to the user more user friendly, avoiding too many key entering during interaction.

Though the improvement in precision and relative recall is not dramatic as we hoped, it nevertheless, tells us that it is wise and helpful to use a spellchecker in a search engine. But will we have the same results if we set up the same experiment but with different types of questionnaires? With a different test panel?

We would propose in the future as an answer to the questions above the following steps in order to discover the full potential of our search engine:

- Run the same experiment but with fewer trials than five and still maintain high score.
- Run the same experiment but with different test panels and compare the results. This, to see how much precision and relative recall would vary. We have notice that our test panel made very few spelling

errors; it would be interesting to run the same experiment with an "error-prone" test panel.
- Introduce more complex words like technical terms and obscure names in the questionnaires to query for.

Acknowledgments

This abstract is part of my master's thesis, and was carried out at the Department of Numerical Analysis and Computing Science (NADA) at the Royal Institute of Technology (KTH). First of all I would like to thank my supervisor, Hercules Dalianis, for providing lot of support, encouragement and advises. I would also like to thank some people who helped a long the way: Johan Carlberger at Euroling AB for providing technical support on the search engine SiteSeeker when needed and Martin Hassel NADA/KTH for his assistance before and during the experimental phase. Finally I thank all the students in language engineering class who took part of the experiment.

References

Carlberger, J., H. Dalianis, M.Hassel and O. Knutsson. Improving Precision in Information Retrieval for Swedish using Stemming. In the Proceedings of NODALIDA '01 - 13th Nordic Conference on Computational Linguistics, May 21-22, 2001, Uppsala, Sweden, 2001.

Carlberger, J. and V. Kann. 2000. Some applications of a statistical tagger for Swedish. Proc. 4:th conference of the International Quantitative Linguistics Association (Qualico-2000), pp. 51-52, August 2000

Carlberger, J. and V. Kann. Implementing an efficient part-of-speech tagger. Software Practice and Experience, 29, pp. 815-832, 1999.

Dalianis, H. Evaluating a Spelling Support in a Search Engine, in Natural Language Processing and Information Systems, 6th International Conference on Applications of Natural Language to Information Systems, NLDB 2002 (Eds.) B. Andersson, M. Bergholtz, P. Johannesson, Stockholm, Sweden, June 27-28, 2002. Lecture Notes in Computer Science. Vol. 2553. pp. 183-190. Springer Verlag, 2002.

Domeij, R., J. Hollman, and Viggo Kann. Detection of spelling errors in Swedish not using a word list en Clair. Journal of Quantitative Linguistics 1:195-201, 1994.

Google press release
Dean, J. (Engineer at Google): Google's future plans. (December 2002).
http://www.searchengineshowdown.com/newsarchive/000611.shtml

Hassel, M. 2001. Internet as Corpus – Automatic Construction of a Swedish News Corpus. NODALIDA '01 - 13th Nordic
Conference on Computational Linguistics, May 21-22 2001, Uppsala, Sweden.

Hodge J., Victoria & Jim Austin. (2001a) An Evaluation of Standard Spell Checking Algorithms and a Binary Neural Approach. Accepted for, IEEE Transactions on Knowledge and Data Engineering

Hodge J. Victoria and Jim Austin. (2001b) An Evaluation of Phonetic Spell Checkers, Technical Report YCS 338(2001), Department of Computer Science, University of York, 2001

Kann, V. , R. Domeij, J. Hollman, M. Tillenius. Implementation aspects and applications of a spelling correction algorithm.
NADA report TRITA-NA-9813, 1998.

Knutsson, O. Automatisk språkgranskning av svensk text (in Swedish), (Automatic Proofreading of Swedish text), Licentiate Thesis. IPLAB-NADA, Royal Institute of Technology, KTH, Stockholm, 2001.

Kukich, K. Techniques for automatically correcting words in text, ACM Computing Surveys. Vol.24, No. 4 (Dec. 1992), pp. 377-439, 1992.

Stolpe, D. 2003 Högre kvalitet med automatisk textbehandling? En utvärdering av SUNETs Webbkatalog (In English Higher Quality by Automatic Text Processing? An Evaluation of the SUNET Web Catalogue, Examensarbete i Datalogi på Kungliga Tekniska Högskolan (forthcoming 2003)
http://www.f.kth.se/~f92-dst/foo.pdf

Webopedia web definition: Search Engine (October 2002)
http://www.webopedia.com/TERM/s/search_engine.html