# Improving Query Spelling Correction Using Web Search Results

**Qing Chen**
Natural Language Processing Lab
Northeastern University
Shenyang, Liaoning, China, 110004
chenqing@ics.neu.edu.cn

**Mu Li**
Microsoft Research Asia
5F Sigma Center
Zhichun Road, Haidian District
Beijing, China, 100080
muli@microsoft.com

**Ming Zhou**
Microsoft Research Asia
5F Sigma Center
Zhichun Road, Haidian District
Beijing, China, 100080
mingzhou@microsoft.com

## Abstract

Traditional research on spelling correction in natural language processing and information retrieval literature mostly relies on pre-defined lexicons to detect spelling errors. But this method does not work well for web query spelling correction, because there is no lexicon that can cover the vast amount of terms occurring across the web. Recent work showed that using search query logs helps to solve this problem to some extent. However, such approaches cannot deal with rarely-used query terms well due to the data sparseness problem. In this paper, a novel method is proposed for use of web search results to improve the existing query spelling correction models solely based on query logs by leveraging the rich information on the web related to the query and its top-ranked candidate. Experiments are performed based on real-world queries randomly sampled from search engine's daily logs, and the results show that our new method can achieve 16.9% relative *F*-measure improvement and 35.4% overall error rate reduction in comparison with the baseline method.

## 1 Introduction

Nowadays more and more people are using Internet search engine to locate information on the web. Search engines take text queries that users type as input, and present users with information of ranked web pages related to users' queries. During this process, one of the important factors that lead to poor search results is misspelled query terms. Actually misspelled queries are rather commonly observed in query logs, as shown in previous investigations into the search engine's log data that around 10%~15% queries were misspelled (Cucerzan and Brill, 2004).

Sometimes misspellings are due to simple typographic errors such as *teh* for *the*. In many cases the spelling errors are more complicated cognitive errors such as *camoflauge* for *camouflage*. As a matter of fact, correct spelling is not always an easy task – even many Americans cannot exactly spell out California governor's last name: *Schwarzenegger*. A spelling correction tool can help improve users' efficiency in the first case, but it is more useful in the latter since the users cannot figure out the correct spelling by themselves.

There has been a long history of general-purpose spelling correction research in natural language processing and information retrieval literature (Kukich, 1992), but its application to web search



*Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 181–189, Prague, June 2007. 

query is still a new challenge. Although there are some similarities in correction candidate generation and selection, these two settings are quite different in one fundamental problem: How to determine the validity of a search term. Traditionally, the measure is mostly based on a pre-defined spelling lexicon – all character strings that cannot be found in the lexicon are judged to be invalid. However, in the web search context, there is little hope that we can construct such a lexicon with ideal coverage of web search terms. For example, even manually collecting a full list of car names and company names will be a formidable task.

To obtain more accurate understanding of this problem, we performed a detailed investigation over one week's MSN daily query logs, among which found that 16.5% of search terms are out of the scope of our spelling lexicon containing around 200,000 entries. In order to get more specific numbers, we also manually labeled a query data set that contains 2,323 randomly sampled queries and 6,318 terms. In this data set, the ratio of out-of-vocabulary (OOV) terms is 17.4%, which is very similar to the overall distribution. However, only 25.3% of these OOV terms are identified to be misspelled, which occupy 85% of the overall spelling errors. All these statistics indicate that accurate OOV term classification is of crucial importance to good query spelling correction performance.

Cucerzan and Brill (2004) first investigated this issue and proposed to use query logs to infer correct spellings of misspelled terms. Their principle can be summarized as follows: given an input query string $q$, finding a more probable query $c$ than $q$ within a confusion set of $q$, in which the edit distance between each element and $q$ is less than a given threshold. They reported good recall for misspelled terms, but without detailed discussions on accurate classification of valid out-of-vocabulary terms and misspellings. In Li's work, distributional similarity metrics estimated from query logs were proposed to be used to discriminate high-frequent spelling errors such as *massenger* from valid out-of-vocabulary terms such as *biocycle*. But this method suffers from the data sparseness problem: sufficient amounts of occurrences of every possible misspelling and valid terms are required to make good estimation of distributional similarity metrics; thus this method does not work well for rarely-used out-of-vocabulary search terms and uncommon misspellings.

In this paper we propose to use web search results to further improve the performance of query spelling correction models. The key contribution of our work is to identify that the dynamic online search results can serve as additional evidence to determine users' intended spelling of a given term. The information in web search results we used includes the number of pages matched for the query, the term distribution in the web page snippets and URLs. We studied two schemes to make use of the returning results of a web search engine. The first one only exploits indicators of the input query's returning results, while the other also looks at other potential correction candidate's search results. We performed extensive evaluations on a query set randomly sampled from search engines' daily query logs, and experimental results show that we can achieve 35.4% overall error rate reduction and 18.2% relative *F*-measure improvement on OOV misspelled terms.

The rest of the paper is structured as follows. Section 2 details other related work of spelling correction research. In section 3, we show the intuitive motivations to use web search results for the query spelling correction. After presenting the formal statement of the query spelling correction problem in Section 4, we describe our approaches that use machine learning methods to integrate statistical features from web search results in Section 5. We present our evaluation methods for the proposed methods and analyze their performance in Section 6. Section 7 concludes the paper.

## 2 Related Work

Spelling correction models in most previous work were constructed based on conventional task settings. Based on the focus of these task settings, two lines of research have been applied to deal with non-word errors and real-word errors respectively.

Non-word error spelling correction is focused on the task of generating and ranking a list of possible spelling corrections for each word not existing in a spelling lexicon. Traditionally candidate ranking is based on manually tuned scores such as assigning alternative weights to different edit operations or leveraging candidate frequencies (Damerau, 1964; Levenshtein, 1966). In recent years, statistical models have been widely used for the tasks of natural language processing, including spelling correction task. (Brill and Moore, 2000) presented an improved error model over the one proposed by (Kernighan et al., 1990) by allowing generic string-to-string edit operations, which helps with modeling major cognitive errors such as the confusion between *le* and *al*. Via explicit modeling of phonetic information of English words, (Toutanova and Moore, 2002) further investigated this issue. Both of them require misspelled/correct word pairs for training, and the latter also needs a pronunciation lexicon, but recently (Ahmad and Kondrak, 2005) demonstrated that it is also possible to learn such models automatically from query logs with the EM algorithm, which is similar to work of (Martin, 2004), learning from a very large corpus of raw text for removing non-word spelling errors in large corpus. All the work for non-word spelling correction focused on the current word itself without taking into account contextual information.

Real-word spelling correction is also referred to be context sensitive spelling correction (CSSC), which tries to detect incorrect usage of valid words in certain contexts. Using a pre-defined confusion set is a common strategy for this task, such as in the work of (Golding and Roth, 1996) and (Mangu and Brill, 1997). Opposite to non-word spelling correction, in this direction only contextual evidences were taken into account for modeling by assuming all spelling similarities are equal.

The complexity of query spelling correction task requires the combination of these types of evidence, as done in (Cucerzan and Brill, 2004; Li et al., 2006). One important contribution of our work is that we use web search results as extended contextual information beyond query strings by taking advantage of application specific knowledge. Although the information used in our methods can all be accessed in a search engine's web archive, such a strategy involves web-scale data processing which is a big engineering challenge, while our method is a light-weight solution to this issue.

## 3 Motivation

When a spelling correction model tries to make a decision whether to make a suggestion $c$ to a query $q$, it generally needs to leverage two types of evidence: the similarity between $c$ and $q$, and the validity plausibility of $c$ and $q$. All the previous work estimated plausibility of a query based on the query string itself – typically it is represented as the string probability, which is further decomposed into production of consecutive n-gram probabilities. For example, both the work of (Cucerzan and Brill, 2004; Li et al., 2006) used n-gram statistical language models trained from search engine's query logs to estimate the query string probability.

In the following, we will show that the search results for a query can serve as a feedback mechanism to provide additional evidences to make better spelling correction decisions. The usefulness of web search results can be two-fold:

First, search results can be used to validate query terms, especially those not popular enough in query logs. One case is the validation for navigational queries (Broder, 2004). Navigational queries usually contain terms that are key parts of destination URLs, which may be out-of-vocabulary terms since there are millions of sites on the web. Because some of these navigational terms are very relatively rare in query logs, without knowledge of the special navigational property of a term, a query spelling correction model might confuse them with other low-frequency misspellings. But such information can be effectively obtained from the URLs of retrieved web pages. Inferring navigational queries through term-URL matching thus can help reduce the chance that the spelling correction model changes an uncommon web site name into popular search term, such as from *innovet* to *innovate*. Another example is that search results can be used in identifying acronyms or other abbreviations. We can observe some clear text patterns that relate abbreviations to their full spellings in the search results as shown in Figure 1. But such mappings cannot easily be obtained from query logs.


Figure 1. Sample search results for *SARS*

Second, search results can help verify correction candidates. The terms appearing in search results, both in the web page titles and snippets, provide additional evidences for users intention. For example, if a user searches for a misspelled query *vaccum cleaner* on a search engine, it is very likely that he will obtain some search results containing the correct term *vacuum* as shown in Figure 2. This can be attributed to the collective link text distribution on the web – many links with misspelled text point to sites with correct spellings. Such evidences can boost the confidence of a spelling correction model to suggest *vacuum* as a correction.

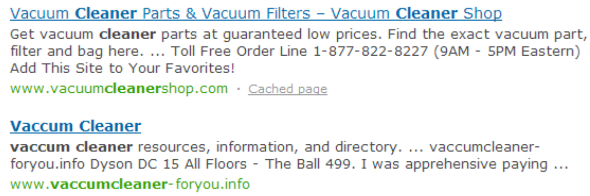


Figure 2. Sample search results for *vaccum cleaner*

The number of matched pages can be used to measure the popularity of a query on the web, which is similar to term frequencies occurring in query logs, but with broader coverage. Poor correction candidates can usually be verified by a smaller number of matched web pages.

Another observation is that the documents retrieved with correctly-spelled query and misspelled ones are similar to some extent in the view of term distribution. Both the web retrieval results of *vacuum* and *vaccum* contain terms such as *cleaner*, *pump*, *bag* or *systems*. We can take this similarity as an evidence to verify the spelling correction results.

## 4 Problem Statement

Given a query $q$, a spelling correction model is to find a query string $c$ that maximizes the posterior probability of $c$ given $q$ within the confusion set of $q$. Formally we can write this as follows:

$$c^* = \mathbf{argmax}_{c \in C} \; Pr(c|q) \quad \textbf{(1)}$$

where $C$ is the confusion set of $q$. Each query string $c$ in the confusion set is a correction candidate for $q$, which satisfies the constraint that the spelling similarity between $c$ and $q$ is within given threshold $\delta$.

In this formulation, the error detection and correction are performed in a unified way. The query $q$ itself always belongs to its confusion set $C$, and when the spelling correction model identifies a more probable query string $c$ in $C$ which is different from $q$, it claims a spelling error detected and makes a correction suggestion $c$.

There are two tasks in this framework. One is how to learn a statistical model to estimate the conditional probability $Pr(c|q)$, and the other is how to generate confusion set $C$ of a given query $q$.

### 4.1 Maximum Entropy Model for Query Spelling Correction

We take a feature-based approach to model the posterior probability $Pr(c|q)$. Specifically we use the maximum entropy model (Berger et al., 1996) for this task:

$$Pr(c|q) = \frac{\exp(\sum_{i=1}^{N} \lambda_i f_i(c,q))}{\sum_c \exp(\sum_{i=1}^{N} \lambda_i f_i(c,q))} \quad \textbf{(2)}$$

where $\sum_c \exp(\sum_{i=1}^{N} \lambda_i f_i(c,q))$ is the normalization factor; $f_i(c,q)$ is a feature function defined over query $q$ and correction candidate $c$, while $\lambda_i$ is the corresponding feature weight. $\lambda s$ can be optimized using the numerical optimization algorithms such as the Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff, 1972) by maximizing the posterior probability of the training set which contains a manually labeled set of query-truth pairs:

$$\lambda^* = \operatorname{argmax} \sum_{c,q} \log Pr_\lambda(c|q) \quad \textbf{(3)}$$

The advantage of maximum entropy model is that it provides a natural way and unified framework to integrate all available information sources. This property is well fit for our task in which we are using a wide variety of evidences based on lexicon, query log and web search results.

### 4.2 Correction Candidate Generation

Correction candidate generation for a query $q$ can be decomposed into two phases. In the first phase, correction candidates are generated for each term in the query from a term-base extracted from query logs. This task can leverage conventional spelling correction methods such as generating candidates based on edit distance (Cucerzan and Brill, 2004) or phonetic similarity (Philips, 1990). Then the correction candidates of the entire query are generated by composing the correction candidates of each individual term. Let $q = w_1 \cdots w_n$, and the confusion set of $w_i$ is $C_{w_i}$, then the confusion set of $q$ is $C_{w_1} \otimes C_{w_2} \otimes \cdots \otimes C_{w_n}$[1]. For example, for a query $q = w_1 w_2$, $w_1$ has candidates $c_{11}$ and $c_{12}$, while $w_2$ has candidates $c_{21}$ and $c_{22}$, then the confusion set $C$ is $\{c_{11}c_{21}, c_{11}c_{22}, c_{12}c_{21}, c_{12}c_{22}\}$.

[1] For denotation simplicity, we do not cover compound and composition errors here.

The problem of this method is the size of confusion set $C$ may be huge for multi-term queries. In practice, one term may have hundreds of possible candidates, then a query containing several terms may have millions. This might lead to impractical search and training using the maximum entropy modeling method. Our solution to this problem is to use *candidate pruning*. We first roughly rank the candidates based on the statistical n-gram language model estimated from query logs. Then we only choose a subset of $C$ that contains a specified number of top-ranked (most probable) candidates to present to the maximum entropy model for offline training and online re-ranking, and the number of candidates is used as a parameter to balance top-line performance and run-time efficiency. This subset can be efficiently generated as shown in (Li et al., 2006).

## 5 Web Search Results based Query Spelling Correction

In this section we will describe in detail the methods for use of web search results in the query spelling correction task. In our work we studied two schemes. The first one only employs indicators of the input query's search results, while the other also looks at the most probable correction candidates' search results. For each scheme, we extract additional scheme-specific features from the available search results, combine them with baseline features and construct a new maximal model to perform candidate ranking.

### 5.1 Baseline model

We denote the maximum entropy model based on baseline model feature set as M0 and the feature set S0 derived from the latest state of the art works of (Li et al., 2006), where S0 includes the features mostly concerning the statistics of the query terms and the similarities between query terms and their correction candidates.

### 5.2 Scheme 1: Using search results for input query only

In this scheme we build more features for each correction candidate (including input query $q$ itself) by distilling more evidence from the search results of the query. S1 denotes the augmented feature set, and M1 denotes the maximum entropy model based on S1. The features are listed as follows:

1. **Number of pages returned**: the number of web search pages retrieved by a web search engine, which is used to estimate the popularity of query. This feature is only for $q$.
2. **URL string**: Binary features indicating whether the combination of terms of each candidate is in the URLs of top retrieved documents. This feature is for all candidates.
3. **Frequency of correction candidate term**: the number of occurrences of modified terms in the correction candidate found in the title and snippet of top retrieved documents based on the observation that correction terms possibly co-occur with their misspelled ones. This feature is invalid for $q$.
4. **Frequency of query term**: the number of occurrences of each term of $q$ found in the title or snippet of the top retrieved documents, based on the observation that the correct terms always appear frequently in their search results.
5. **Abbreviation pattern**: Binary features indicating whether inputted query terms might be abbreviations according to text patterns in search results.

### 5.3 Scheme 2: Using both search results of input query and top-ranked candidate

In this scheme we extend the use of search results both for query $q$ and for top-ranked candidate $c$ other than $q$ determined by M1. First we submit a query to a search engine for the initial retrieval to obtain one set of search results $R_q$, then use M1 to find the best correction candidate $c$ other than $q$. Next we perform a second retrieval with $c$ to obtain another set of search results $R_c$. Finally additional features are generated for each candidate based on $R_c$, then a new maximum entropy model M2 is built to re-rank the candidates for a second time. The entire process can be schematically shown in Figure 3.

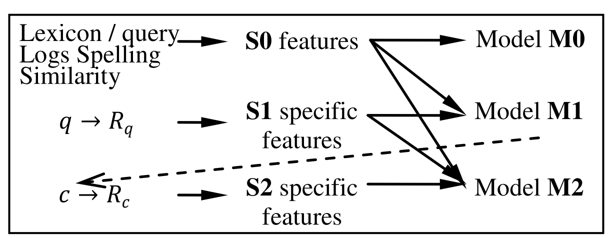


Figure 3. Relations of models and features

where $R_q$ is the web search results of query $q$; $R_c$ is the web search results of $c$ which is the top-ranked correction of $q$ suggested by model M1.

The new feature set denoted with S2 is a set of document similarities between $R_q$ and $R_c$, which includes different similarity estimations between the query and its correction at the document level using merely cosine measure based on term frequency vectors of $R_q$ and $R_c$.

# 6 Experiments

## 6.1 Evaluation Metrics

In our work, we consider the following four types of evaluation metrics:

- **Accuracy**: The number of correct outputs proposed by the spelling correction model divided by the total number of queries in the test set
- **Recall**: The number of correct suggestions for misspelled queries by the spelling correction model divided by the total number of misspelled queries in the test set
- **Precision**: The number of correct suggestions for misspelled queries proposed by the spelling correction model divided by the total number of suggestions made by the system
- **F-measure**: Formula $F = 2PR/(P + R)$ used for calculating the f-measure, which is essentially the harmonic mean of recall and precision

Any individual metric above might not be sufficient to indicate the overall performance of a query spelling correction model. For example, as in most retrieval tasks, we can trade recall for precision or vice versa. Although intuitively $F$ might be in accordance with accuracy, there is no strict theoretical relation between these two numbers – there are conditions under which accuracy improves while $F$-measure may drop or be unchanged.

## 6.2 Experimental Setup

We used a manually constructed data set as gold standard for evaluation. First we randomly sampled 7,000 queries from search engine's daily query logs of different time periods, and had them manually labeled by two annotators independently. Each query is attached to a truth, which is either the query itself for valid queries, or a spelling correction for misspelled ones. From the annotation results that both annotators agreed with each other, we extracted 2,323 query-truth pairs as training set and 991 as test set. Table 1 shows the statistics of the data sets, in which $E_q$ denotes the error rate of query and $E_t$ denotes the error rate of term.

| | # queries | # terms | $E_q$ | $E_t$ |
|---|---|---|---|---|
| Training set | 2,323 | 6,318 | 15.0% | 5.6% |
| Test set | 991 | 2,589 | 12.8% | 5.2% |

Table 1. Statistics of training set and test set

In the following experiments, at most 50 correction candidates were used in the maximum entropy model for each query if there is no special explanation. The web search results were fetched from MSN's search engine. By default, top 100 retrieved items from the web retrieval results were used to perform feature extraction. A set of query log data spanning 9 months are used for collecting statistics required by the baseline.

## 6.3 Overall Results

Following the method as described in previous sections, we first ran a group of experiments to evaluate the performance of each model we discussed with default settings. The detailed results are shown in Table 2.

| Model | Accuracy | Recall | Precision | F |
|---|---|---|---|---|
| M0 | 91.8% | 60.6% | 62.6% | 0.616 |
| M1 | 93.9% | 64.6% | 77.4% | 0.704 |
| M2 | 94.7% | 66.9% | 78.0% | 0.720 |

Table 2. Overall Results

From the table we can observe significant performance boosts on all evaluation metrics of M1 and M2 over M0.

We can achieve 25.6% error rate reduction and 23.6% improvement in precision, as well as 6.6% relative improvement in recall, when adding S1 to M1. Paired t-test gives $p$-value of 0.002, which is significant to 0.01 level.

M2 can bring additional 13.1% error rate reduction and moderate improvement in precision, as well as 3.6% improvement in recall over M1, with paired t-test showing that the improvement is significant to 0.01 level.

### 6.4 Impact of Candidate number

Theoretically the number of correction candidates in the confusion set determines the accuracy and recall upper bounds for all models concerned in this paper. Performance might be hurt if we use a too small candidate number, which is because the corrections are separated from the confusion sets. These curves shown in Figure 4 and 5, include both theoretical bound (oracle) and actual performance of our described models. From the chart we can see that our models perform best when $N_t$ is around 50, and when $N_t > 15$ the oracle recall and accuracy almost stay unchanged, thus the actual models' performance only benefits a little from larger $N_t$ values. The missing part of recall is largely due to the fact that we are not able to generate truth candidates for some weird query terms rather than insufficient size of confusion set.

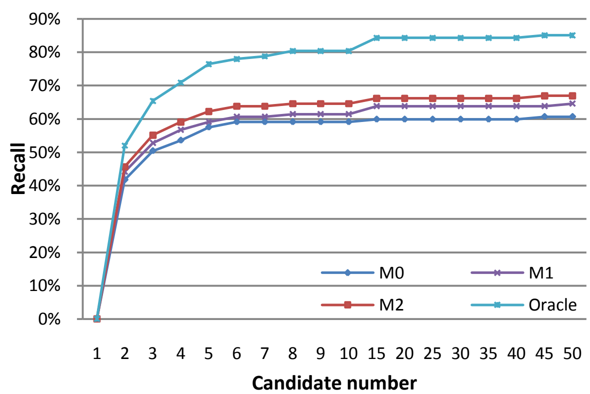


Figure 4. Recall versus candidate number

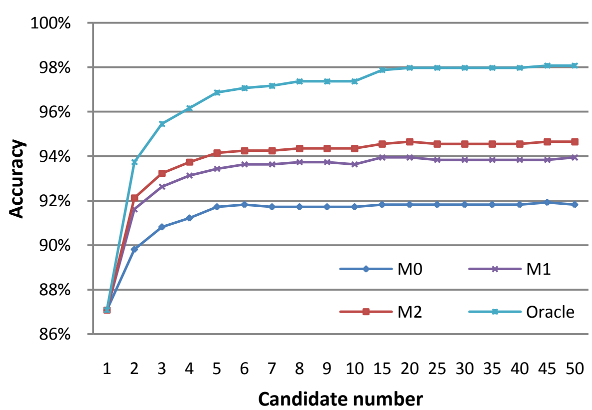


Figure 5. Accuracy versus candidate number

### 6.5 Discussions

We also studied the performance difference between in-vocabulary (IV) and out-of-vocabulary (OOV) terms when using different spelling correction models. The detailed results are shown in Table 3 and Table 4.

| | Accuracy | Precision | Recall | F |
|---|---|---|---|---|
| **M0** | 88.2% | 77.1% | 67.3% | 0.718 |
| **M1** | 92.4% | 88.5% | 77.3% | 0.825 |
| **M2** | 93.2% | 91.6% | 79.1% | 0.849 |

Table 3. OOV Term Results

| | Accuracy | Precision | Recall | F |
|---|---|---|---|---|
| **M0** | 98.8% | 44.0% | 45.8% | 0.449 |
| **M1** | 99.0% | 62.5% | 20.8% | 0.313 |
| **M2** | 99.1% | 75.0% | 37.5% | 0.500 |

Table 4. IV Term Results

The results show that M1 is very powerful to identify and correct OOV spelling errors compared with M0. Actually M1 is able to correct spelling errors such as *guiness*, whose frequency in query log is even higher than its truth spelling *guinness*. Since most spelling errors are OOV terms, this explains why the model M1 can significantly outperform the baseline. But for IV terms things are different. Although the overall accuracy is better, the F-measure of M1 is far lower than M0. M2 performs best for the IV task in terms of both accuracy and F-measure. However, IV spelling errors is so small a portion of the total misspelling (only 17.4% of total spelling errors in our test set) that the room for improvement is very small. This helps to explain why the performance gap between M1 and M0 is much larger than the one between M2 and M1, and shows the tendency that M1 prefer to identify and correct OOV misspellings in comparison to IV ones, which causes F-measure drop from M0 to M1; while by introducing more useful evidence, M2 outperforms better for both OOV and IV terms over M0 and M1.

Another set of statistics we collected from the experiments is the performance data of low-frequency terms when using the models proposed in this paper, since we believe that our approach would help make better classification of low-frequency search terms. As a case study, we identified in the test set all terms whose frequencies in our query logs are less than 800, and for these terms we calculated the error reduction rate of model M1 over the baseline model M0 at each interval of 50. The detailed results are shown in Figure 6. The clear trend can be observed that M1 can achieve larger error rate reduction over baseline for terms with lower frequencies. This is because the performance of baseline model drops for these terms when there are no reliable distributional similarity estimations available due to data sparseness in query logs, while M1 can use web data to alleviate this problem.

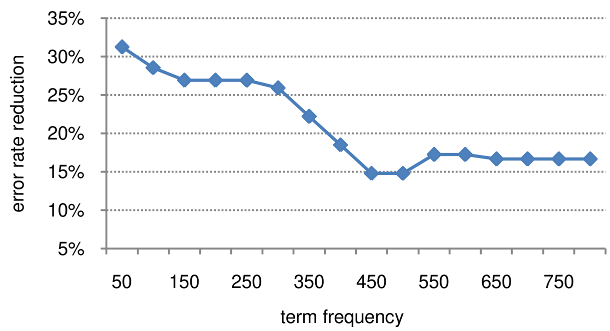


Figure 6. Error rate reduction of M1 over baseline for terms in different frequency ranges

## 7 Conclusions and Future Work

The task of query spelling correction is very different from conventional spelling checkers, and poses special research challenges. In this paper, we presented a novel method for use of web search results to improve existing query spelling correction models.

We explored two schemes for taking advantage of the information extracted from web search results. Experimental results show that our proposed methods can achieve statistically significant improvements over the baseline model which only relies on evidences of lexicon, spelling similarity and statistics estimated from query logs.

There is still further potential useful information that should be studied in this direction. For example, we can work on page ranking information of returning pages, because trusted or well-known sites with high page rank generally contain few wrong spellings. In addition, the term co-occurrence statistics on the returned snippet text are also worth deep investigation.

## References

Ahmad F. and Grzegorz Kondrak G. Learning a spelling error model from search query logs. *Proceedings of EMNLP 2005*, pages 955-962, 2005

Berger A. L., Della Pietra S. A., and Della Pietra V. J. A maximum entropy approach to natural language processing. *Computation Linguistics*, 22(1):39-72, 1996

Brill E. and Moore R. C. An improved error model for noisy channel spelling correction. *Proceedings of 38th annual meeting of the ACL*, pages 286-293, 2000.

Broder, A. A taxonomy of web search. *SIGIR Forum Fall 2002*, Volume 36 Number 2, 2002.

Church K. W. and Gale W. A. Probability scoring for spelling correction. In *Statistics and Computing*, volume 1, pages 93-103, 1991.

Cucerzan S. and Brill E. Spelling correction as an iterative process that exploits the collective knowledge of web users. *Proceedings of EMNLP'04*, pages 293-300, 2004.

Damerau F. A technique for computer detection and correction of spelling errors. *Communication of the ACM* 7(3):659-664, 1964.

Darroch J. N. and Ratcliff D. Generalized iterative scaling for long-linear models. *Annals of Mathematical Statistics*, 43:1470-1480, 1972.

Efthimiadis, N.E., Query Expansion, In *Annual Review of Information Systems and Technology*, Vol. 31, pp. 121-187 , 1996.

Golding A. R. and Roth D. Applying winnow to context-sensitive spelling correction. *Proceedings of ICML 1996*, pages 182-190, 1996.

J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'2001*, pages 111-119, Sept 2001.

J. Xu and W. Croft. Query expansion using local and global document analysis. In *Proceedings of the SIGIR 1996*, pages 4-11, 1996

Kernighan M. D., Church K. W. and Gale W. A. A spelling correction program based on a noisy channel model. *Proceedings of COLING 1990*, pages 205-210, 1990.

Kukich K. Techniques for automatically correcting words in text. *ACM Computing Surveys*. 24(4): 377-439, 1992.

Levenshtein V. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physice – Doklady* 10: 707-710, 1966.

Li M., Zhu M. H., Zhang Y. and Zhou M. Exploring distributional similarity based models for query spelling correction. *Proceedings of COLING-ACL 2006*, pages 1025-1032, 2006

Mangu L. and Eric Brill E. Automatic rule acquisition for spelling correction. *Proceedings of ICML 1997*, pages 734-741, 1997.

Martin Reynaert. Text induced spelling correction. *Proceedings of COLING 2004,*pages 834-840, 2004.

Mayes E., Damerau F. and Mercer R. Context based spelling correction. *Information processing and management,* 27(5): 517-522, 1991.

Philips L. Hanging on the metaphone. *Computer Language Magazine*, 7(12): 39, 1990.

Toutanova K. and Moore R. Pronunciation modeling for improved spelling correction. *Proceedings of the 40th annual meeting of ACL*, pages 144-151, 2002.