Exploring Distributional Similarity Based Models for Query Spelling Correction

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

Muhua Zhu
School of
Information Science and Engineering
Northeastern University
Shenyang, Liaoning, China, 110004
zhumh@ics.neu.edu.cn

Yang Zhang
School of
Computer Science and Technology
Tianjin University
Tianjin, China, 300072
yangzhang@tju.edu.cn

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

Abstract

A query speller is crucial to search engine in improving web search relevance. This paper describes novel methods for use of distributional similarity estimated from query logs in learning improved query spelling correction models. The key to our methods is the property of distributional similarity between two terms: it is high between a frequently occurring misspelling and its correction, and low between two irrelevant terms only with similar spellings. We present two models that are able to take advantage of this property. Experimental results demonstrate that the distributional similarity based models can significantly outperform their baseline systems in the web query spelling correction task.

1 Introduction

Investigations into query log data reveal that more than 10% of queries sent to search engines contain misspelled terms (Cucerzan and Brill, 2004). Such statistics indicate that a good query speller is crucial to search engine in improving web search relevance, because there is little opportunity that a search engine can retrieve many relevant contents with misspelled terms.

The problem of designing a spelling correction program for web search queries, however, poses special technical challenges and cannot be well solved by general purpose spelling correction methods. Cucerzan and Brill (2004) discussed in detail specialties and difficulties of a query spell checker, and illustrated why the existing methods could not work for query spelling correction. They also identified that no single evidence, either a conventional spelling lexicon or term frequency in the query logs, can serve as criteria for validate queries.

To address these challenges, we concentrate on the problem of learning improved query spelling correction model by integrating distributional similarity information automatically derived from query logs. The key contribution of our work is identifying that we can successfully use the evidence of distributional similarity to achieve better spelling correction accuracy. We present two methods that are able to take advantage of distributional similarity information. The first method extends a string edit-based error model with confusion probabilities within a generative source channel model. The second method explores the effectiveness of our approach within a discriminative maximum entropy model framework by integrating distributional similarity-based features. Experimental results demonstrate that both methods can significantly outperform their baseline systems in the spelling correction task for web search queries.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 1025–1032,
Sydney, July 2006.

The rest of the paper is structured as follows: after a brief overview of the related work in Section 2, we discuss the motivations for our approach, and describe two methods that can make use of distributional similarity information in Section 3. Experiments and results are presented in Section 4. The last section contains summaries and outlines promising future work.

2 Related Work

The method for web query spelling correction proposed by Cucerzan and Brill (2004) is essentially based on a source channel model, but it requires iterative running to derive suggestions for very-difficult-to-correct spelling errors. Word bigram model trained from search query logs is used as the source model, and the error model is approximated by inverse weighted edit distance of a correction candidate from its original term. The weights of edit operations are interactively optimized based on statistics from the query logs. They observed that an edit distance-based error model only has less impact on the overall accuracy than the source model. The paper reports that un-weighted edit distance will cause the overall accuracy of their speller's output to drop by around 2%. The work of Ahmad and Kondrak (2005) tried to employ an unsupervised approach to error model estimation. They designed an EM (Expectation Maximization) algorithm to optimize the probabilities of edit operations over a set of search queries from the query logs, by exploiting the fact that there are more than 10% misspelled queries scattered throughout the query logs. Their method is concerned with single character edit operations, and evaluation was performed on an isolated word spelling correction task.

There are two lines of research in conventional spelling correction, which deal with non-word errors and real-word errors respectively. Non-word error spelling correction is concerned with the task of generating and ranking a list of possible spelling corrections for each query word not found in a lexicon. While traditionally candidate ranking is based on manually tuned scores such as assigning weights to different edit operations or leveraging candidate frequencies, some statistical models have been proposed for this ranking task in recent years. Brill and Moore (2000) presented an improved error model over the one proposed by Kernigham 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*. Toutanova and Moore (2002) further explored this via explicit modeling of phonetic information of English words. Both these two methods require misspelled/correct word pairs for training, and the latter also needs a pronunciation lexicon. Real-word spelling correction is also referred to as context sensitive spelling correction, which tries to detect incorrect usage of valid words in certain contexts (Golding and Roth, 1996; Mangu and Brill, 1997).

Distributional similarity between words has been investigated and successfully applied in many natural language tasks such as automatic semantic knowledge acquisition (Dekang Lin, 1998) and language model smoothing (Essen and Steinbiss, 1992; Dagan et al., 1997). An investigation on distributional similarity functions can be found in (Lillian Lee, 1999).

3 Distributional Similarity-Based Models for Query Spelling Correction

3.1 Motivation

Most of the previous work on spelling correction concentrates on the problem of designing better error models based on properties of character strings. This direction ever evolves from simple Damerau-Levenshtein distance (Damerau, 1964; Levenshtein, 1966) to probabilistic models that estimate string edit probabilities from corpus (Church and Gale, 1991; Mayes et al, 1991; Ristad and Yianilos, 1997; Brill and Moore, 2000; and Ahmad and Kondrak, 2005). In the mentioned methods, however, the similarities between two strings are modeled on the average of many misspelling-correction pairs, which may cause many idiosyncratic spelling errors to be ignored. Some of those are typical word-level cognitive errors. For instance, given the query term *adventura*, a character string-based error model usually assigns similar similarities to its two most probable corrections *adventure* and *aventura*. Taking into account that *adventure* has a much higher frequency of occurring, it is most likely that *adventure* would be generated as a suggestion. However, our observation into the query logs reveals that *adventura* in most cases is actually a common misspelling of *aventura*. Two annotators were asked to judge 36 randomly sampled queries that contain more than one term, and they agreed upon that 35 of them should be *aventura*.

To solve this problem, we consider alternative methods to make use of the information beyond a term's character strings. Distributional similarity provides such a dimension to view the possibility that one word can be replaced by another based on the statistics of words co-occuring with them. Distributional similarity has been proposed to perform tasks such as language model smoothing and word clustering, but to the best of our knowledge, it has not been explored in estimating similarities between misspellings and their corrections. In this section, we will only involve the consine metric for illustration purpose.

Query logs can serve as an excellent corpus for distributional similarity estimation. This is because query logs are not only an up-to-date term base, but also a comprehensive spelling error repository (Cucerzan and Brill, 2004; Ahmad and Kondrak, 2005). Given enough size of query logs, some misspellings, such as *adventura*, will occur so frequently that we can obtain reliable statistics of their typical usage. Essential to our method is the observation of high distributional similarity between frequently occurring spelling errors and their corrections, but low between irrelevant terms. For example, we observe that *adventura* occurred more than 3,300 times in a set of logged queries that spanned three months, and its context was similar to that of *aventura*. Both of them usually appeared after words like *peurto* and *lyrics*, and were followed by *mall*, *palace* and *resort*. Further computation shows that, in the *tf* (term frequency) vector space based on surrounding words, the cosine value between them is approximately 0.8, which indicates these two terms are used in a very similar way among all the users trying to search *aventura*. The cosine between *adventura* and *adventure* is less than 0.03 and basically we can conclude that they are two irrelevant terms, although their spellings are similar.

Distributional similarity is also helpful to address another challenge for query spelling correction: differentiating valid OOV terms from frequently occurring misspellings.

	InLex	Freq	Cosine
vaccum	No	18,430	0.99
vacuum	Yes	158,428	
seraphin	No	1,718	0.30
seraphim	Yes	14,407	

Table 1. Statistics of two word pairs with similar spellings

Table 1 lists detailed statistics of two word pairs, each of pair of words have similar spelling, lexicon and frequency properties. But the distributional similarity between each pair of words provides the necessary information to make correction classification that *vacuum* is a spelling error while *seraphin* is a valid OOV term.

3.2 Problem Formulation

In this work, we view the query spelling correction task as a statistical sequence inference problem. Under the probabilistic model framework, it can be conceptually formulated as follows. Given a correction candidate set *C for a* query string *q*:

$$C = \{c \mid EditDist(q,c) < \delta\}$$

in which each correction candidate *c* satisfies the constraint that the edit distance between *c* and *q* is less than a given threshold *δ,* the model is to find *c** in *C* with the highest probability:

$$c^* = \arg\max_{c \in C} P(c \mid q) \qquad (1)$$

In practice, the correction candidate set *C* is not generated from the entire query string directly. Correction candidates are generated for each term of a query first, and then *C* is constructed by composing the candidates of individual terms. The edit distance threshold *δ* is set for each term proportionally to the length of the term.

3.3 Source Channel Model

Source channel model has been widely used for spelling correction (Kernigham et al., 1990; Mayes, Damerau et al., 1991; Brill and More, 2000; Ahmad and Kondrak, 2005). Instead of directly optimize (1), source channel model tries to solve an equivalent problem by applying Bayes's rule and dropping the constant denominator:

$$c^* = \arg\max_{c \in C} P(q \mid c)P(c) \qquad (2)$$

In this approach, two component generative models are involved: source model *P*(*c*) that generates the user's intended query *c* and error model *P*(*q*|*c*) that generates the real query *q* given *c*. These two component models can be independently estimated.

In practice, for a multi-term query, the source model can be approximated with an n-gram statistical language model, which is estimated with tokenized query logs. Taking bigram model for example, *c* is a correction candidate containing *n* terms, $c = c_1c_2 \ldots c_n$, then *P*(*c*) can be written as the product of consecutive bigram probabilities:

$$P(c) = \prod P(c_i \mid c_{i-1})$$

Similarly, the error model probability of a query is decomposed into generation probabilities of individual terms which are assumed to be independently generated:

$$P(q \mid c) = \prod P(q_i \mid c_i)$$

Previous proposed methods for error model estimation are all based on the similarity between the character strings of q_i and c_i as described in 3.1. Here we describe a distributional similarity-based method for this problem. Essentially there are different ways to estimate distributional similarity between two words (Dagan et al., 1997), and the one we propose to use is *confusion probability* (Essen and Steinbiss, 1992). Formally, confusion probability P_c estimates the possibility that one word w_1 can be replaced by another word w_2:

$$P_c(w_2 \mid w_1) = \sum_w \frac{P(w \mid w_1)}{P(w)} P(w \mid w_2) P(w_2) \quad (3)$$

where w belongs to the set of words that co-occur with both w_1 and w_2.

From the spelling correction point of view, given w_1 to be a valid word and w_2 one of its spelling errors, $P_c(w_2 \mid w_1)$ actually estimates opportunity that w_1 is misspelled as w_2 in query logs. Compared to other similarity measures such as cosine or Euclidean distance, confusion probability is of interest because it defines a probabilistic distribution rather than a generic measure. This property makes it more theoretically sound to be used as error model probability in the Bayesian framework of the source channel model. Thus it can be applied and evaluated independently. However, before using confusion probability as our error model, we have to solve two problems: probability renormalization and smoothing.

Unlike string edit-based error models, which distribute a major portion of probability over terms with similar spellings, confusion probability distributes probability over the entire vocabulary in the training data. This property may cause the problem of unfair comparison between different correction candidates if we directly use (3) as the error model probability. This is because the synonyms of different candidates may share different portion of confusion probabilities. This problem can be solved by re-normalizing the probabilities only over a term's possible correction candidates and itself. To obtain better estimation, here we also require that the frequency of a correction candidate should be higher than that of the query term, based on the observation that correct spellings generally occur more often in query logs. Formally, given a word w and its correction candidate set C, the confusion probability of a word w' conditioned on w can be redefined as

$$P_c(w' \mid w) = \begin{cases} \dfrac{P'_c(w' \mid w)}{\sum_{c \in C} P'_c(c \mid w)} & w' \in C \\ 0 & w' \notin C \end{cases} \quad (4)$$

where $P'_c(w' \mid w)$ is the original definition of confusion probability.

In addition, we might also have the zero-probability problem when the query term has not appeared or there are few context words for it in the query logs. In such cases there is no distributional similarity information available to any known terms. To solve this problem, we define the final error model probability as the linear combination of confusion probability and a string edit-based error model probability $P_{ed}(q \mid c)$:

$$P(q \mid c) = \lambda P_c(q \mid c) + (1 - \lambda) P_{ed}(q \mid c) \quad (5)$$

where λ is the interpolation parameter between 0 and 1 that can be experimentally optimized on a development data set.

3.4 Maximum Entropy Model

Theoretically we are more interested in building a unified probabilistic spelling correction model that is able to leverage all available features, which could include (but not limited to) traditional character string-based typographical similarity, phonetic similarity and distributional similarity proposed in this work. The maximum entropy model (Berger et al., 1996) provides us with a well-founded framework for this purpose, which has been extensively used in natural lan guage processing tasks ranging from part-of-speech tagging to machine translation.

For our task, the maximum entropy model defines a posterior probabilistic distribution $P(c \mid q)$ over a set of feature functions f_i (q, c) defined on an input query q and its correction candidate c:

$$P(c \mid q) = \frac{\exp \sum_{i=1}^{N} \lambda_i f_i(c,q)}{\sum_c \exp \sum_{i=1}^{N} \lambda_i f_i(c,q)} \quad (6)$$

where λs are feature weights, which can be optimized by maximizing the posterior probability on the training set:

$$\lambda^* = \arg\max_{\lambda} \sum_{(t,q)\in TD} \log P_{\lambda}(t \mid q)$$

where *TD* denotes the set of training samples in the form of query-truth pairs presented to the training algorithm.

We use the Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff, 1972) to learn the model parameter λs of the maximum entropy model. GIS training requires normalization over all possible prediction classes as shown in the denominator in equation (6). Since the potential number of correction candidates may be huge for multi-term queries, it would not be practical to perform the normalization over the entire search space. Instead, we use a method to approximate the sum over the *n*-best list (a list of most probable correction candidates). This is similar to what Och and Ney (2002) used for their maximum entropy-based statistical machine translation training.

3.4.1 Features

Features used in our maximum entropy model are classified into two categories I) baseline features and II) features supported by distributional similarity evidence. Below we list the feature templates.

Category I:

1. *Language model probability feature*. This is the only real-valued feature with feature value set to the logarithm of source model probability:

$$f_{prob}(q,c) = \log P(c)$$

2. *Edit distance-based features*, which are generated by checking whether the weighted Levenshtein edit distance between a query term and its correction is in certain range;

All the following features, including this one, are binary features, and have the feature function of the following form:

$$f_n(q,c) = \begin{cases} 1 & \textit{constraint satisfied} \\ 0 & \textit{otherwise} \end{cases}$$

in which the feature value is set to 1 when the constraints described in the template are satisfied; otherwise the feature value is set to 0.

3. *Frequency-based features*, which are generated by checking whether the frequencies of a query term and its correction candidate are above certain thresholds;

4. *Lexicon-based features*, which are generated by checking whether a query term and its correction candidate are in a conventional spelling lexicon;

5. *Phonetic similarity-based features*, which are generated by checking whether the edit distance between the metaphones (Philips, 1990) of a query term and its correction candidate is below certain thresholds.

Category II:

6. *Distributional similarity based term features*, which are generated by checking whether a query term's frequency is higher than certain thresholds but there are no candidates for it with higher frequency and high enough distributional similarity. This is usually an indicator that the query term is valid and not covered by the spelling lexicon. The frequency thresholds are enumerated from 10,000 to 50,000 with the interval 5,000.

7. *Distributional similarity based correction candidate features*, which are generated by checking whether a correction candidate's frequency is higher than the query term or the correction candidate is in the lexicon, and at the same time the distributional similarity is higher than certain thresholds. This generally gives the evidence that the query term may be a common misspelling of the current candidate. The distributional similarity thresholds are enumerated from 0.6 to 1 with the interval 0.1.

4 Experimental Results

4.1 Dataset

We randomly sampled 7,000 queries from daily query logs of MSN Search and they were manually labeled by two annotators. For each query identified to contain spelling errors, corrections were given by the annotators independently. From the annotation results that both annotators agreed upon 3,061 queries were extracted, which were further divided into a test set containing 1,031 queries and a training set containing 2,030 queries. In the test set there are 171 queries identified containing spelling errors with an error rate of 16.6%. The numbers on the training set is 312 and 15.3%, respectively. The average length of queries on training set is 2.8 terms and on test set it is 2.6.

In our experiments, a term bigram model is used as the source model. The bigram model is trained with query log data of MSN Search during the period from October 2004 to June 2005. Correction candidates are generated from a term base extracted from the same set of query logs.

For each of the experiments, the performance is evaluated by the following metrics:

Accuracy: The number of correct outputs generated by the system divided by the total number of queries in the test set;

Recall: The number of correct suggestions for misspelled queries generated by the system divided by the total number of misspelled queries in the test set;

Precision: The number of correct suggestions for misspelled queries generated by the system divided by the total number of suggestions made by the system.

4.2 Results

We first investigated the impact of the interpolation parameter λ in equation (5) by applying the confusion probability-based error model on training set. For the string edit-based error model probability $P_{ed}(q\,|\,c)$, we used a heuristic score computed as the inverse of weighted edit distance, which is similar to the one used by Cucerzan and Brill (2004).

Figure 1 shows the accuracy metric at different settings of λ. The accuracy generally gains improvements before λ reaches 0.9. This shows that confusion probability plays a more important role in the combination. As a result, we empirically set λ= 0.9 in the following experiments.

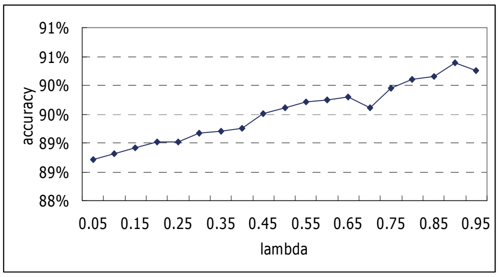

Figure 1. Accuracy with different λs

To evaluate whether the distributional similarity can contribute to performance improvements, we conducted the following experiments. For source channel model, we compared the confusion probability-based error model (SC-SimCM) against two baseline error model settings, which are source model only (SC-NoCM) and the heuristic string edit-based error model (SC-EdCM) we just described. Two maximum entropy models were trained with different feature sets. ME-NoSim is the model trained only with baseline features. It serves as the baseline for ME-Full, which is trained with all the features described in 3.4.1. In training ME-Full, cosine distance is used as the similarity measure examined by feature functions.

In all the experiments we used the standard viterbi algorithm to search for the best output of source channel model. The *n*-best list for maximum entropy model training and testing is generated based on language model scores of correction candidates, which can be easily obtained by running the forward-viterbi backward-A* algorithm. On a 3.0GHZ Pentium4 personal computer, the system can process 110 queries per second for source channel model and 86 queries per second for maximum entropy model, in which 20 best correction candidates are used.

Model	Accuracy	Recall	Precision
SC-NoCM	79.7%	63.3%	40.2%
SC-EdCM	84.1%	62.7%	47.4%
SC-SimCM	88.2%	57.4%	58.8%
ME-NoSim	87.8%	52.0%	60.0%
ME-Full	89.0%	60.4%	62.6%

Table 2. Performance results for different models

Table 2 details the performance scores for the experiments, which shows that both of the two distributional similarity-based models boost accuracy over their baseline settings. SC-SimCM achieves 26.3% reduction in error rate over SC-EdCM, which is significant to the 0.001 level (paired t-test). ME-Full outperforms ME-NoSim in all three evaluation measures, with 9.8% reduction in error rate and 16.2% improvement in recall, which is significant to the 0.01 level.

It is interesting to note that the accuracy of SC-SimCM is slightly better than ME-NoSim, although ME-NoSim makes use of a rich set of features. ME-NoSim tends to keep queries with frequently misspelled terms unchanged (e.g. *caffine extractions from soda*) to reduce false alarms (e.g. *bicycle* suggested for *biocycle*).

We also investigated the performance of the models discussed above at different recall. Figure 2 and Figure 3 show the precision-recall curves and accuracy-recall curves of different models. We observed that the performance of SC-SimCM and ME-NoSim are very close to each other and ME-Full consistently yields better performance over the entire P-R curve.

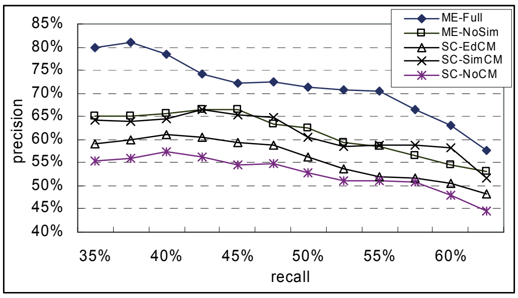

Figure 2. Precision-recall curve of different models

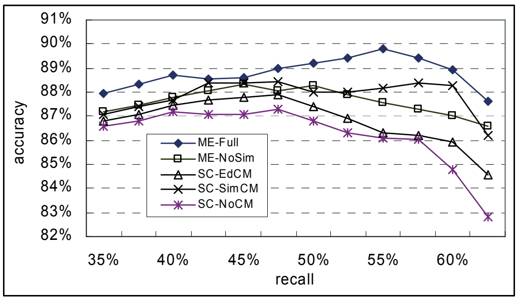

Figure 3. Accuracy-recall curve of different models

We performed a study on the impact of training size to ensure all models are trained with enough data.

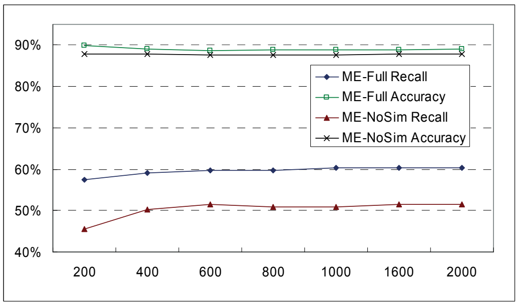

Figure 4. Accuracy of maximum entropy models trained with different number of samples

Figure 4 shows the accuracy of the two maximum entropy models as functions of number of training samples. From the results we can see that after the number of training samples reaches 600 there are only subtle changes in accuracy and recall. Therefore basically it can be concluded that 2,000 samples are sufficient to train a maximum entropy model with the current feature sets.

5 Conclusions and Future Work

We have presented novel methods to learn better statistical models for the query spelling correction task by exploiting distributional similarity information. We explained the motivation of our methods with the statistical evidence distilled from query log data. To evaluate our proposed methods, two probabilistic models that can take advantage of such information are investigated. Experimental results show that both methods can achieve significant improvements over their baseline settings.

A subject of future research is exploring more effective ways to utilize distributional similarity even beyond query logs. Currently for low-frequency terms in query logs there are no reliable distribution similarity evidence available for them. A promising method of dealing with this in next steps is to explore information in the resulting page of a search engine, since the snippets in the resulting page can provide far greater detailed information about terms in a query.

References

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

Adam L. Beger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computation Linguistics,* 22(1):39-72.

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

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

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

Ido Dagan, Lillian Lee and Fernando Pereira. 1997. Similarity-Based Methods for Word Sense Disambiguation. *Proceedings of the 35th annual meeting of ACL*, pages 56-63.

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

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

Ute Essen and Volker Steinbiss. 1992. Co-occurrence smoothing for stochastic language modeling. *Proceedings of ICASSP*, volume 1, pages 161-164.

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

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

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

Lillian Lee. 1999. Measures of distributional similarity. *Proceedings of the 37th annual meeting of ACL*, pages 25-32.

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

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. *Proceedings of COLING-ACL 1998*, pages 768-774.

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

Eric Mayes, Fred Damerau and Robert Mercer. 1991. Context based spelling correction. *Information processing and management* 27(5): 517-522.

Franz Och and Hermann Ney. 2002. Discriminative training and maimum entropy models for statistical machine translation. *Proceedings of the 40th annual meeting of ACL*, pages 295-302.

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

Eric S. Ristad and Peter N. Yianilos. 1997. Learning string edit distance. *Proceedings of ICML 1997*. pages 287-295

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