Contextual Spelling Correction Using Latent Semantic Analysis

Michael P. Jones and **James H. Martin**
Dept. of Computer Science and Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0430
{mjones,martin}@cs.colorado.edu

Abstract

Contextual spelling errors are defined as the use of an incorrect, though valid, word in a particular sentence or context. Traditional spelling checkers flag misspelled words, but they do not typically attempt to identify words that are used incorrectly in a sentence. We explore the use of *Latent Semantic Analysis* for correcting these incorrectly used words and the results are compared to earlier work based on a Bayesian classifier.

1 Introduction

Spelling checkers are now available for all major word processing systems. However, these spelling checkers only catch errors that result in misspelled words. If an error results in a different, but incorrect word, it will go undetected. For example, *quite* may easily be mistyped as *quiet*. Another type of error occurs when a writer simply doesn't know which word of a set of homophones[1] (or near homophones) is the proper one for a particular context. For example, the usage of *affect* and *effect* is commonly confused.

Though the cause is different for the two types of errors, we can treat them similarly by examining the contexts in which they appear. Consequently, no effort is made to distinguish between the two error types and both are called contextual spelling errors. Kukich (1992a; 1992b) reports that 40% to 45% of observed spelling errors are contextual errors. Sets of words which are frequently misused or mistyped for one another are identified as confusion sets. Thus, from our earlier examples, {*quiet, quite*} and {*affect, effect*} are two separate confusion sets.

In this paper, we introduce Latent Semantic Analysis (LSA) as a method for correcting contextual spelling errors for a given collection of confusion sets. LSA was originally developed as a model for information retrieval (Dumais et al., 1988; Deerwester et al., 1990), but it has proven useful in other tasks too. Some examples include an expert Expert locator (Streeter and Lochbaum, 1988) and a conference proceedings indexer (Foltz, 1995) which performs better than a simple keyword-based index. Recently, LSA has been proposed as a theory of semantic learning (Landauer and Dumais, (In press)).

Our motivation in using LSA was to test its effectiveness at predicting words based on a given sentence and to compare it to a Bayesian classifier. LSA makes predictions by building a high-dimensional, "semantic" space which is used to compare the similarity of the words from a confusion set to a given context. The experimental results from LSA prediction are then compared to both a baseline predictor and a hybrid predictor based on trigrams and a Bayesian classifier.

2 Related Work

Latent Semantic Analysis has been applied to the problem of spelling correction previously (Kukich, 1992b). However, this work focused on detecting misspelled words, not contextual spelling errors. The approach taken used letter n-grams to build the semantic space. In this work, we use the words directly.

Yarowsky (1994) notes that conceptual spelling correction is part of a closely related class of problems which include word sense disambiguation, word choice selection in machine translation, and accent and capitalization restoration. This class of problems has been attacked by many others. A number of feature-based methods have been tried, including Bayesian classifiers (Gale, Church, and Yarowsky, 1992; Golding, 1995), decision lists (Yarowsky, 1994), and knowledge-based approaches (McRoy, 1992). Recently, Golding and Schabes (1996) described a system, Tribayes, that combines a trigram model of the words' parts of speech with a Bayesian classifier. The trigram component of the system is used to make decisions for those confusion sets that

[1] Homophones are words that sound the same, but are spelled differently.

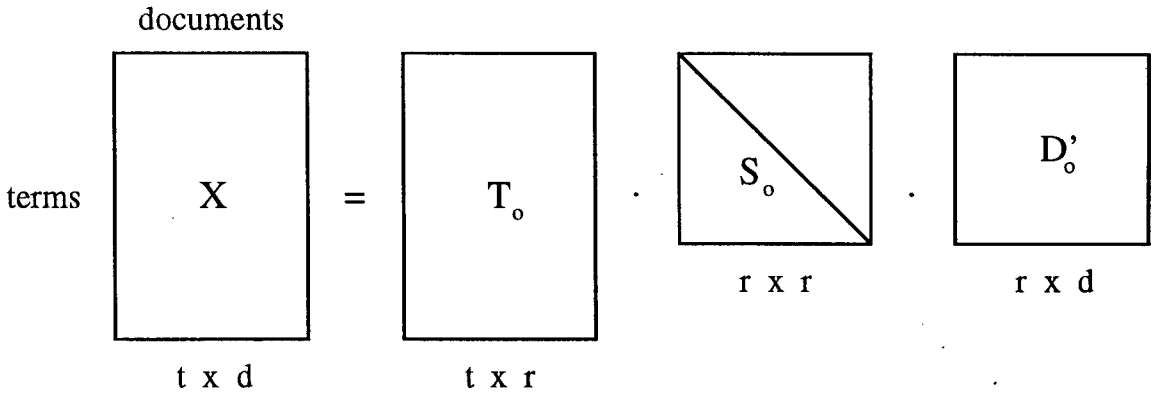

Figure 1: Singular value decomposition (SVD) of matrix X produces matrices T, S and D'.

contain words with different parts of speech. The Bayesian component is used to predict the correct word from among same part-of-speech words.

Golding and Schabes selected 18 confusion sets from a list of commonly confused words plus a few that represent typographical errors. They trained their system using a random 80% of the Brown corpus (Kučera and Francis, 1967). The remaining 20% of the corpus was used to test how well the system performed. We have chosen to use the same 18 confusion sets and the Brown corpus in order to compare LSA to Tribayes.

3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) was developed at Bellcore for use in information retrieval tasks (for which it is also known as LSI) (Dumais et al., 1988; Deerwester et al., 1990). The premise of the LSA model is that an author begins with some idea or information to be communicated. The selection of particular lexical items in a collection of texts is simply evidence for the underlying ideas or information being presented. The goal of LSA, then, is to take the "evidence" (i.e., words) presented and uncover the underlying semantics of the text passage. Because many words are polysemous (have multiple meanings) and synonymous (have meanings in common with other words), the evidence available in the text tends to be somewhat "noisy." LSA attempts to eliminate the noise from the data by first representing the texts in a high-dimensional space and then reducing the dimensionality of the space to only the most important dimensions. This process is described in more detail in Dumais (1988) or Deerwester (1990), but a brief description is provided here.

A collection of texts is represented in matrix format. The rows of the matrix correspond to terms and the columns represent documents. The individual cell values are based on some function of the term's frequency in the corresponding document and its frequency in the whole collection. The function for selecting cell values will be discussed in section 4.2. A singular value decomposition (SVD) is performed on this matrix. SVD factors the original matrix into the product of three matrices. We'll identify these matrices as T, S, and D' (see Figure 1). The T matrix is a representation of the original term vectors as vectors of derived orthogonal factor values. D' is a similar representation for the original document vectors. S is a diagonal matrix[2] of rank r. It is also called the singular value matrix. The singular values are sorted in decreasing order along the diagonal. They represent a scaling factor for each dimension in the T and D' matrices.

Multiplying T, S, and D' together perfectly reproduces the original representation of the text collection. Recall, however, that the original representation is expected to be noisy. What we really want is an approximation of the original space that eliminates the majority of the noise and captures the most important ideas or semantics of the texts.

An approximation of the original matrix is created by eliminating some number of the least important singular values in S. They correspond to the least important (and hopefully, most noisy) dimensions in the space. This step leaves a new matrix (S_o) of rank k.[3] A similar reduction is made in T and D by retaining the first k columns of T and the first k rows of D' as depicted in Figure 2. The product of the resulting T_o, S_o, and D'_o matrices is a least squares best fit reconstruction of the original matrix (Eckart and Young, 1939). The reconstructed matrix defines a space that represents or predicts the frequency with which each term in the space would appear in a given document or text segment given an infinite sample of semantically similar texts (Lan-

[2] A diagonal matrix is a square matrix that contains non-zero values only along the diagonal running from the upper left to the lower right.

[3] The number of factors k to be retained is generally selected empirically.

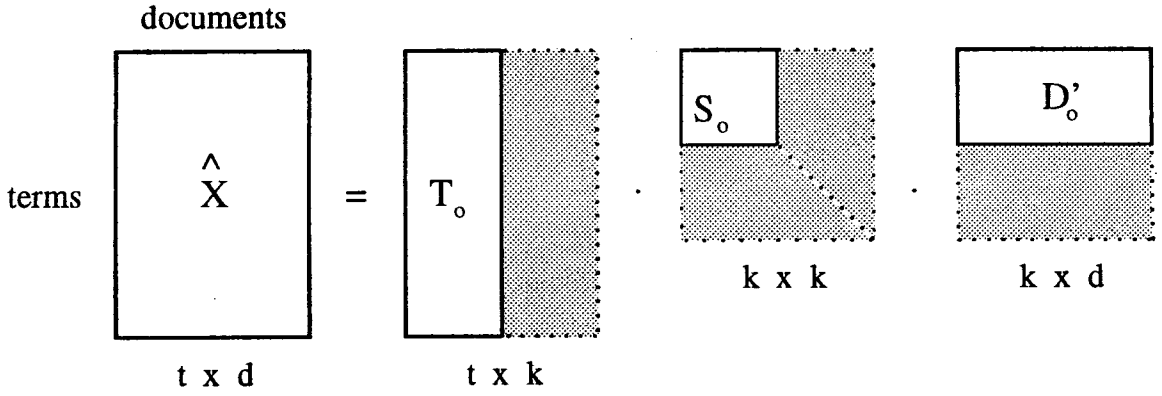

Figure 2: Results of reducing the T, S and D' matrices produced by SVD to rank k. Recombining the reduced matrices gives $\hat{X}$, a least squares best fit reconstruction of the original matrix.

dauer and Dumais, (In press)).

New text passages can be projected into the space by computing a weighted average of the term vectors which correspond to the words in the new text. In the contextual spelling correction task, we can generate a vector representation for each text passage in which a confusion word appears. The similarity between this text passage vector and the confusion word vectors can be used to predict the most likely word given the context or text in which it will appear.

4 Experimental Method

4.1 Data

Separate corpora for training and testing LSA's ability to correct contextual word usage errors were created from the Brown corpus (Kučera and Francis, 1967). The Brown corpus was parsed into individual sentences which are randomly assigned to either a training corpus or a test corpus. Roughly 80% of the original corpus was assigned as the training corpus and the other 20% was reserved as the test corpus. For each confusion set, only those sentences in the training corpus which contained words in the confusion set were extracted for construction of an LSA space. Similarly, the sentences used to test the LSA space's predictions were those extracted from the test corpus which contained words from the confusion set being examined. The details of the space construction and testing method are described below.

4.2 Training

Training the system consists of processing the training sentences and constructing an LSA space from them. LSA requires the corpus to be segmented into documents. For a given confusion set, an LSA space is constructed by treating each training sentence as a document. In other words, each training sentence is used as a column in the LSA matrix. Before being processed by LSA, each sentence undergoes the following transformations: context reduction, stemming, bigram creation, and term weighting.

Context reduction is a step in which the sentence is reduced in size to the confusion word plus the seven words on either side of the word or up to the sentence boundary. The average sentence length in the corpus is 28 words, so this step has the effect of reducing the size of the data to approximately half the original. Intuitively, the reduction ought to improve performance by disallowing the distantly located words in long sentences to have any influence on the prediction of the confusion word because they usually have little or nothing to do with the selection of the proper word. In practice, however, the reduction we use had little effect on the predictions obtained from the LSA space.

We ran some experiments in which we built LSA spaces using the whole sentence as well as other context window sizes. Smaller context sizes didn't seem to contain enough information to produce good predictions. Larger context sizes (up to the size of the entire sentence) produced results which were not significantly different from the results reported here. However, using a smaller context size reduces the total number of unique terms by an average of 13%. Correspondingly, using fewer terms in the initial matrix reduces the average running time and storage space requirements by 17% and 10% respectively.

Stemming is the process of reducing each word to its morphological root. The goal is to treat the different morphological variants of a word as the same entity. For example, the words *smile*, *smiled*, *smiles*, *smiling*, and *smilingly* (all from the corpus) are reduced to the root *smile* and treated equally. We tried different stemming algorithms and all improved the predictive performance of LSA. The results presented in this paper are based on Porter's (Porter, 1980) algorithm.

Bigram creation is performed for the words that were not removed in the context reduction step.

Bigrams are formed between all adjacent pairs of words. The bigrams are treated as additional terms during the LSA space construction process. In other words, the bigrams fill their own row in the LSA matrix.

Term weighting is an effort to increase the weight or importance of certain terms in the high dimensional space. A local and global weighting is given to each term in each sentence. The local weight is a combination of the raw count of the particular term in the sentence and the term's proximity to the confusion word. Terms located nearer to the confusion word are given additional weight in a linearly decreasing manner. The local weight of each term is then flattened by taking its $\log_2$. The global weight given to each term is an attempt to measure its predictive power in the corpus as a whole. We found that entropy (see also (Lochbaum and Streeter, 1989)) performed best as a global measure. Furthermore, terms which did not appear in more than one sentence in the training corpus were removed.

While LSA can be used to quickly obtain satisfactory results, some tuning of the parameters involved can improve its performance. For example, we chose (somewhat arbitrarily) to retain 100 factors for each LSA space. We wanted to fix this variable for all confusion sets and this number gives a good average performance. However, tuning the number of factors to select the "best" number for each space shows an average of 2% improvement over all the results and up to 8% for some confusion sets.

4.3 Testing

Once the LSA space for a confusion set has been created, it can be used to predict the word (from the confusion set) most likely to appear in a given sentence. We tested the predictive accuracy of the LSA space in the following manner. A sentence from the test corpus is selected and the location of the confusion word in the sentence is treated as an unknown word which must be predicted. One at a time, the words from the confusion set are inserted into the sentence at the location of the word to be predicted and the same transformations that the training sentences undergo are applied to the test sentence. The inserted confusion word is then removed from the sentence (but not the bigrams of which it is a part) because its presence biases the comparison which occurs later. A vector in LSA space is constructed from the resulting terms.

The word predicted most likely to appear in a sentence is determined by comparing the similarity of each test sentence vector to each confusion word vector from the LSA space. Vector similarity is evaluated by computing the cosine between two vectors. The pair of sentence and confusion word vectors with the largest cosine is identified and the corresponding confusion word is chosen as the most likely word for the test sentence. The predicted word is compared to the correct word and a tally of correct predictions is kept.

5 Results

The results described in this section are based on the 18 confusion sets selected by Golding (1995; 1996). Seven of the 18 confusion sets contain words that are all the same part of speech and the remaining 11 contain words with different parts of speech. Golding and Schabes (1996) have already shown that using a trigram model to predict words from a confusion set based on the expected part of speech is very effective. Consequently, we will focus most of our attention on the seven confusion sets containing words of the same part of speech. These seven sets are listed first in all of our tables and figures. We also show the results for the remaining 11 confusion sets for comparison purposes, but as expected, these aren't as good. We, therefore, consider our system complementary to one (such as Tribayes) that predicts based on part of speech when possible.

5.1 Baseline Prediction System

We describe our results in terms of a baseline prediction system that ignores the context contained in the test sentence and always predicts the confusion word that occurred most frequently in the training corpus. Table 1 shows the performance of this baseline predictor. The left half of the table lists the various confusion sets. The next two columns show the training and testing corpus sentence counts for each confusion set. Because the sentences in the Brown corpus are not tagged with a markup language, we identified individual sentences automatically based on a small set of heuristics. Consequently, our sentence counts for the various confusion sets differ slightly from the counts reported in (Golding and Schabes, 1996).

The right half of Table 1 shows the most frequent word in the training corpus from each confusion set. Following the most frequent word is the baseline performance data. Baseline performance is the percentage of correct predictions made by choosing the given (most frequent) word. The percentage of correct predictions also represents the frequency of sentences in the test corpus that contain the given word. The final column lists the training corpus frequency of the given word. The difference between the baseline performance column and the training corpus frequency column gives some indication about how evenly distributed the words are between the two corpora.

For example, there are 158 training sentences for the confusion set {*principal, principle*} and 34 test sentences. Since the word *principle* is listed in the right half of the table, it must have appeared more frequently in the training set. From the final column,

Confusion Set	Train	Test	Most Freq.	Base	(Train Freq.)
principal principle	158	34	principle	41.2	(57.6)
raise rise	117	36	rise	72.2	(65.0)
affect effect	193	53	effect	88.7	(85.0)
peace piece	257	62	peace	58.1	(59.5)
country county	389	91	country	59.3	(71.0)
amount number	480	122	number	75.4	(73.8)
among between	853	203	between	62.1	(66.7)
accept except	189	62	except	67.7	(73.5)
begin being	623	161	being	88.8	(89.4)
lead led	197	63	led	50.8	(52.3)
passed past	353	81	past	64.2	(63.2)
quiet quite	280	76	quite	88.2	(76.1)
weather whether	267	67	whether	73.1	(79.0)
cite sight site	128	32	sight	62.5	(54.7)
it's its	1577	391	its	84.7	(84.9)
than then	2497	578	than	58.8	(55.3)
you're your	734	220	your	86.8	(84.5)
their there they're	4176	978	there	53.4	(53.1)

Table 1: Baseline performance for 18 confusion sets. The table is divided into confusion sets containing words of the same part of speech and those which have different parts of speech.

we can see that it occurred in almost 58% of the training sentences. However, it occurs in only 41% of the test sentences and thus the baseline predictor scores only 41% for this confusion set.

5.2 Latent Semantic Analysis

Table 2 shows the performance of LSA on the contextual spelling correction task. The table provides the baseline performance information for comparison to LSA. In all but the case of {*amount, number*}, LSA improves upon the baseline performance. The improvement provided by LSA averaged over all confusion sets is about 14% and for the sets with the same part of speech, the average improvement is 16%.

Table 2 also gives the results obtained by Tribayes as reported in (Golding and Schabes, 1996). The baseline performance given in connection with Tribayes corresponds to the partitioning of the Brown corpus used to test Tribayes. It should be noted that we did not implement Tribayes nor did we use the same partitioning of the Brown corpus as Tribayes. Thus, the comparison between LSA and Tribayes is an indirect one.

The differences in the baseline predictor for each system are a result of different partitions of the Brown corpus. Both systems randomly split the data such that roughly 80% is allocated to the training corpus and the remaining 20% is reserved for the test corpus. Due to the random nature of this process, however, the corpora must differ between the two systems. The baseline predictor presented in this paper and in (Golding and Schabes, 1996) are based on the same method so the corresponding columns in Table 2 can be compared to get an idea of the distribution of sentences that contain the most frequent word for each confusion set.

Examination of Table 2 reveals that it is difficult to make a direct comparison between the results of LSA and Tribayes due to the differences in the partitioning of the Brown corpus. Each system should perform well on the most frequent confusion word in the training data. Thus, the distribution of the most frequent word between the the training and the test corpus will affect the performance of the system. Because the baseline score captures information about the percentage of the test corpus that should be easily predicted (i.e., the portion that contains the most frequent word), we propose a comparison of the results by examination of the respective systems' improvement over the baseline score reported for each. The results of this comparison are charted in Figure 3. The horizontal axis in the figure represents the baseline predictor performance for each system (even though it varies between the two systems). The vertical bar thus represents the performance above (or below) the baseline predictor for each system on each confusion set.

LSA performs slightly better, on average, than Tribayes for those confusion sets which contain words of the same part of speech. Tribayes clearly out-performs LSA for those words of a different part of speech. Thus, LSA is doing better than the Bayesian component of Tribayes, but it doesn't include part of speech information and is therefore not capable of performing as well as the part of speech trigram component of Tribayes. Consequently, we believe that LSA is a competitive alternative to

Confusion Set	LSA		Tribayes	
	Baseline	LSA	Baseline	Tribayes
principal principle	41.2	91.2	58.8	88.2
raise rise	72.2	80.6	64.1	76.9
affect effect	88.7	94.3	91.8	95.9
peace piece	58.1	83.9	44.0	90.0
country county	59.3	81.3	91.9	85.5
amount number	75.4	56.6	71.5	82.9
among between	62.1	80.8	71.5	75.3
accept except	67.7	82.3	70.0	82.0
begin being	88.8	93.2	93.2	97.3
lead led	50.8	73.0	46.9	83.7
passed past	64.2	80.3	68.9	95.9
quiet quite	88.2	90.8	83.3	95.5
weather whether	73.1	85.1	86.9	93.4
cite sight site	62.5	78.1	64.7	70.6
it's its	84.7	92.8	91.3	98.1
than then	58.8	90.5	63.4	94.9
you're your	86.8	91.4	89.3	98.9
their there they're	53.4	73.9	56.8	97.6

Table 2: LSA performance for 18 confusion sets. The results of Tribayes (Golding and Schabes, 1996) are also given.

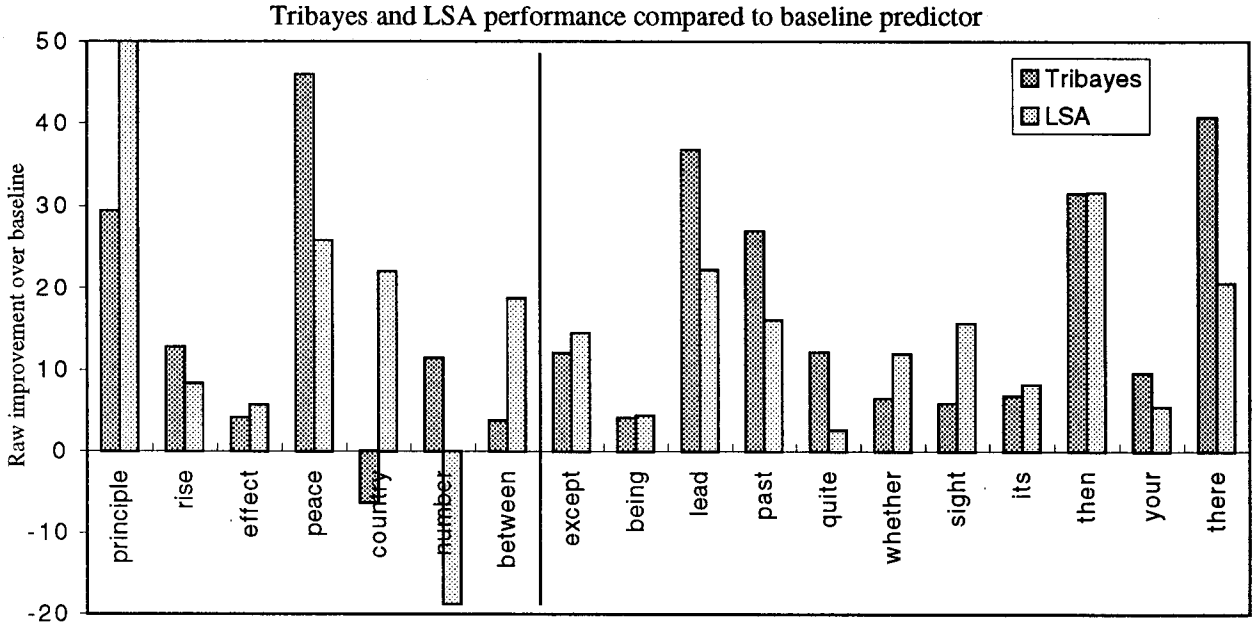

Figure 3: Comparison of Tribayes vs. LSA performance above the baseline metric.

a Bayesian classifier for making predictions among words of the same part of speech.

5.3 Performance Tuning

The results that have been presented here are based on uniform treatment for each confusion set. That is, the initial data processing steps and LSA space construction parameters have all been the same. However, the model does not require equivalent treatment of all confusion sets. In theory, we should be able to increase the performance for each confusion set by tuning the various parameters for each confusion set.

In order to explore this idea further, we selected the confusion set {*amount, number*} as a testbed for performance tuning to a particular confusion set. As previously mentioned, we can tune the number of factors to a particular confusion set. In the case of this confusion set, using 120 factors increases the performance by 6%. However, tuning this parameter alone still leaves the performance short of the baseline predictor.

A quick examination of the context in which both words appear reveals that a significant percentage (82%) of all training instances contain either the bigram of the confusion word preceded by *the*, followed by *of*, or in some cases, both. For example, there are many instances of the collocation *the+number+of* in the training data. However, there are only one third as many training instances for *amount* (the less frequent word) as there are for *number*. This situation leads LSA to believe that the bigrams *the+amount* and *amount+of* have more discrimination power than the corresponding bigrams which contain *number*. As a result, LSA gives them a higher weight and LSA almost always predicts *amount* when the confusion word in the test sentence appears in this context. This local context is a poor predictor of the confusion word and its presence tends to dominate the decision made by LSA. By eliminating the words *the* and *of* from the training and testing process, we permit the remaining context to be used for prediction. The elimination of the poor local context combined with the larger number of factors increases the performance of LSA to 13% above the baseline predictor (compared to 11% for Tribayes). This is a net increase in performance of 32%!

6 Conclusion

We've shown that LSA can be used to attack the problem of identifying contextual misuses of words, particularly when those words are the same part of speech. It has proven to be an effective alternative to Bayesian classifiers. Confusions sets whose words are different parts of speech are more effectively handled using a method which incorporates the word's part of speech as a feature. We are exploring techniques for introducing part of speech information into the LSA space so that the system can make better predictions for those sets on which it doesn't yet measure up to Tribayes. We've also shown that for the cost of experimentation with different parameter combinations, LSA's performance can be tuned for individual confusion sets.

While the results of this experiment look very nice, they still don't tell us anything about how useful the technique is when applied to unedited text. The testing procedure assumes that a confusion word must be predicted as if the author of the text hadn't supplied a word or that writers misuse the confusion words nearly 50% of the time. For example, consider the case of the confusion set {*principal, principle*}. The LSA prediction accuracy for this set is 91%. However, it might be the case that, in practice, people tend to use the correct word 95% of the time. LSA has thus introduced a 4% error into the writing process. Our continuing work is to explore the error rate that occurs in unedited text as a means of assessing the "true" performance of contextual spelling correction systems.

7 Acknowledgments

The first author is supported under DARPA contract SOL BAA95-10. We gratefully acknowledge the comments and suggestions of Thomas Landauer and the anonymous reviewers.

References

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, September.

Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. 1988. Using Latent Semantic Analysis to improve access to textual information. In *Human Factors in Computing Systems, CHI'88 Conference Proceedings (Washington, D.C.)*, pages 281–285, New York, May. ACM.

Carl Eckart and Gale Young. 1939. A principle axis transformation for non-hermitian matrices. *American Mathematical Society Bulletin*, 45:118–121.

Peter W. Foltz. 1995. Improving human-proceedings interaction: Indexing the CHI index. In *Human Factors in Computing Systems: CHI'95 Conference Companion*, pages 101–102. Associations for Computing Machinery (ACM), May.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5-6):415–439, Dec.

Andrew R. Golding. 1995. A Bayesian hybrid method for context-sensitive spelling correction. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA.

Andrew R. Golding and Yves Schabes. 1996. Combining trigram-based and feature-based methods for context-sensitive spelling correction. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Clara, CA, June. Association for Computational Linguistics.

Karen Kukich. 1992a. Spelling correction for the telecommunications network for the deaf. *Communications of the ACM*, 35(5):80–90, May.

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

Henry Kučera and W. Nelson Francis. 1967. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI.

Thomas K. Landauer and Susan T. Dumais. (In press). A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*.

Karen E. Lochbaum and Lynn A. Streeter. 1989. Comparing and combining the effectiveness of Latent Semantic Indexing and the ordinary vector space model for information retrieval. *Information Processing & Management*, 25(6):665–676.

Susan W. McRoy. 1992. Using multiple knowledge sources for word sense disambiguation. *Computational Linguistics*, 18(1):1–30, March.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137, July.

Lynn A. Streeter and Karen E. Lochbaum. 1988. An expert/expert-locating system based on automatic representation of semantic structure. In *Proceedings of the Fourth Conference on Artificial Intelligence Applications*, pages 345–350, San Diego, CA, March. IEEE.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88–95, Las Cruces, NM, June. Association for Computational Linguistics.