# CSE 256 (Spring 2004)
# "Language Models for Spelling Correction"

**Dustin Boswell**
*dboswell [at] cs ucsd edu*

## Abstract

This project examines the use of language models in a spelling correction system that adopts the "Noisy Channel Model". Various models based on bigram counts are tested in an experiment where typos are introduced into a test corpus, and corrections are made by language model ranking alone. Simple bigram models perform noticeably better than the unigram model (84% accuracy vs. 74%). And more sophisticated models perform marginally better (up to 86%).

## 1 Introduction

*Language Modeling* is the field of creating models for written text so that we can assign a probability to a given string of words. For example, the string "he went home" should have a higher probability than "abacus kindly flew". A typical application is in voice recognition, where a language model can help the system rank a set of candidate sentences by how likely they would have been said.

## 2 Using Language Models in Spelling Correction

For this project, we will adopt the "Noisy Channel Model" of spelling correction [MDM91]. Let $\widetilde{S}$ be a sentence of words that appears in some text. We wish to find the most likely sentence $S$ that was *intended* by the writer. ( $\widetilde{S}$ is the result of possible typographical and spelling errors on $S$. ) To do this, we must have an *error model* (how typos and spelling mistakes occur), and a *language model* (how likely sentences are to be intended in the first place). Formally, we wish to find the intended $S^*$ that has the highest likelihood given the observed sentence $\widetilde{S}$:

$$\begin{aligned} S^* &= arg\max_S P(S|\widetilde{S}) \\ &= arg\max_S \frac{P(\widetilde{S}|S) \cdot P(S)}{P(\widetilde{S})} \\ &= arg\max_S P(\widetilde{S}|S) \cdot P(S) \end{aligned}$$

The term $P(\widetilde{S}|S)$ is the error model, and $P(S)$ is the language model that is the focus of this project.

## 3 Previous Work

The earliest work on spelling correction is [Dam64], [Pet80], [PZ84], and [Pet86]. Most early work focused on correcting non-words (as opposed to errors that happen to also be a word), and used simple "hand-coded" rules to do so. Later work including [MDM91], [KCG90], [BM00], and [Tou02] have all used the noisy channel model given above.

## 4 Defining our Language Models

We assume a fixed window of $K$ words $w_1 \cdots w_k \cdots w_K$, where $w_k$ is the "word of focus" - typically the center word. A practical example might be $K = 9$ and $k = 5$. In application, $w_k$ would be the word "currently under suspicion", and the system would be computing $P(w_1 \cdots \hat{w_k} \cdots w_K)$ for all candidate replacements $\hat{w_k}$. Also, the luxury of having forward context ($k < K$) isn't necessary,

although one might imagine it would help, and for applications like document spelling correction it isn't unreasonable to use it.

We tested 7 language models of varying complexity. The following table is a summary. After describing each in more detail, we describe the experiment, and then give performance results.

| Model Name | Window Size | Description |
|---|---|---|
| Null | 0 | All sentences assigned equal probability |
| Unigram | 1 | The prob of a sentence is just the prob of the focus word |
| Before-Bigram | 2 | The bigram prob of the focus word and the word immediately before it |
| After-Bigram | 2 | The bigram prob of the focus word and the word immediately after it |
| Spaced-Bigrams | K | Combines the distance-dependent bigram of every word with the focus word |
| Bag Of Bigrams | K | Combines the distance-averaged bigram of every word with the focus word |
| Hybrid Bigrams | K | A hybrid of Spaced-Bigrams and Bag Of Bigrams |

Figure 1: A Summary of the language models tested in this project.

### 4.1 Null Language Model

This model is simply $P(w_1 \cdots w_k \cdots w_K) \equiv 1 \quad \forall$ sentences. (Note that none of the models need to be true probability functions since they are simply used to *rank* alternatives.) We include this model and measure its performance to get a "baseline" of how difficult the task is. Since ties are broken randomly, using this model results in a system that simply picks a candidate replacement at random.

### 4.2 Unigram Language Model

This model is defined by

$$P(w_1 \cdots w_k \cdots w_K) \equiv P(w_k) = \frac{count(w_k)}{N}$$

That is, it ignores all context and simply returns the probability of the focus word. $count(w_k)$ is the number of times the word $w_k$ occurred in the corpus and $N$ is the size of the corpus. Again, this model is not a true probability function either since the sum over all sentences is much greater than 1. Using this model results in a system that picks the most frequent word of the candidate replacements.

### 4.3 Before Bigram Language Model

This model includes the previous word as context:

$$P(w_1 \cdots w_k \cdots w_K) \equiv P(w_{k-1}w_k) = \frac{count(w_{k-1}w_k)}{N}$$

where $count(w_{k-1}w_k)$ is the number of times that bigram appeared in the corpus.

### 4.4 After Bigram Language Model

This model includes the following word as context:

$$P(w_1 \cdots w_k \cdots w_K) \equiv P(w_k w_{k+1}) = \frac{count(w_k w_{k+1})}{N}$$

where $count(w_k w_{k+1})$ is the number of times that bigram appeared in the corpus.

The rest of the language models are more complex, and use all $K$ words of context. We start by conditioning the probability on $w_k$:

$$P(w_1 \cdots w_k \cdots w_K) \quad = \quad P(w_1 \cdots w_{k-1} w_{k+1} \cdots w_K | w_k) * P(w_k)$$

Now technically our notation is sloppy: a probability should have both a random variable and its value. We will show both from here on, as it will cause confusion later on if we don't. We say $W_i$ is the random variable "which word is in the $i^{th}$ slot of the $K$-word window?". $w_i$ is the particular word that $W_i$ is taking on. To repeat:

$$P(W_1 = w_1, \cdots, W_k = w_k, \cdots, W_K = w_K) = \\ P(W_1 = w_1, \cdots, W_{k-1} = w_{k-1}, W_{k+1} = w_{k+1}, \cdots, W_K = w_K | W_k = w_k) * P(W_k = w_k)$$

Now, we make the *Naive Bayes Assumption*:

$$P(W_1 = w_1, \cdots, W_{k-1} = w_{k-1}, W_{k+1} = w_{k+1}, \cdots, W_K = w_K | W_k = w_k) \quad = \quad \prod_{i \neq k} P(W_i = w_i | W_k = w_k)$$

Notice that this is *not* the "Bag of Words" assumption. The locational distance between the words ($|k - i|$) is still there.

We now propose our first model that uses this directly.

### 4.5 The Spaced-Bigram Model

The model is simply:

$$P(W_1 = w_1 \cdots, W_k = w_k \cdots, W_K = w_K) = P(W_k = w_k) * \prod_{i \neq k} P(W_i = w_i | W_k = w_k)$$

where we compute

$$P(W_k = w_k) = \frac{c(w_k)}{N}$$

as just a simple unigram count estimated from a training corpus and

$$P(W_i = w_i | W_k = w_k) = \frac{P(W_i = w_i, W_k = w_k)}{P(W_k = w_k)} = \frac{c(w_i, w_k, k - i)}{c(w_k)}$$

uses a distance-sensitive bigram count. For example, $c($ `United , America , 3` $)$ would be the number of times `United` appeared and `America` appeared 3 words later in the corpus. $c($ `America , United , -3` $)$ is the same count.

This model has the advantage that it will learn different statistics for adjacent words ( $c(\cdot, \cdot, 1)$ - which should capture syntax rules ), and words further apart ( $c(\cdot, \cdot, d >> 1)$ - which should capture things like sentence topic). However, the parameter space is effectively $K$ times larger than a traditional bigram, and this model may suffer from under-training.

All models we propose may benefit from smoothing and other techniques to adjust the estimated probabilities. Indeed, in the experiments we use Laplace Smoothing.

Notice the case of $K = 3$ effectively becomes the application of traditional bigrams, using a window of size 3.

### 4.6 The Bag-of-Bigrams Model

The Bag of Words assumption is:

$$\begin{aligned} P(W_i = w | W_k = w_k) &= P(W_j = w | W_k = w_k) \qquad \forall i, j \\ &= P(W = w | W_k = w_k) \end{aligned}$$

That is, $P(W = w | W_k = w_k)$ just means "if I pick a random location in the $K$-word window around $w_k$, what's the probability the word at that location is $w$?". And that's why we can drop the unused index $i$. $W$ is simply "a random location in the $K$-word window (other than $k$)". Now, the model is simply:

$$\begin{aligned} P(W_1 = w_1, \cdots, W_k = w_k, \cdots, W_K = w_K) &= P(W_k = w_k) * \prod_{i \neq k} P(W_i = w_i | W_k = w_k) \\ &= P(W_k = w_k) * \prod_{i \neq k} P(W = w_i | W_k = w_k) \end{aligned}$$

where again

$$P(W_k = w_k) = \frac{c(w_k)}{N}$$

but this time

$$P(W = w_i | W_k = w_k) = \frac{P(W = w_i, W_k = w_k)}{P(W_k = w_k)} = \frac{\frac{1}{K-1} \sum_{j \neq k} c(w_i, w_k, k-j)}{c(w_k)}$$

Here, we are counting over all possible locations in windows of size $K$. In essence, $P(W = w_i | W_k = w_k)$ is just an average of $P(W_j = w_i | W_k = w_k)$ over all $j$.

This model has a parameter space that is $K$ times smaller than the previous model, so its probability estimates should be more robust. Unfortunately, this model has no notion of syntax or word order. For instance, if you randomly permute the words (other than $w_k$), the model will assign the same probability to all such permutations.

### 4.7 The Hybrid Model

Our main concern with the Spaced-Bigram model was that the parameter space was large. However, we dislike the Bag-of-Bigram's method of treating adjacent words the same as words that were 5 apart. We propose a hybrid model that treats adjacent words specially, and words that are further than 1 away all the same.

The Hybrid Model makes a "partial" Bag of Words assumption. For locations $i$ in the window such that $|k - i| > 1$, it assumes they are a bag of words

$$\begin{aligned} P(W_i = w | W_k = w_k) = P(W_j = w | W_k = w_k) \qquad & \forall i, j \ : \ \mathbf{|k - i| > 1 \ \ and \ \ |k - j| > 1} \\ &= P_{far}(W = w | W_k = w_k) \end{aligned}$$

Our model is:

$$\begin{aligned} & P(W_1 = w_1, \cdots, W_k = w_k, \cdots, W_K = w_K) \\ & \quad = P(W_k = w_k) * \prod_{i \neq k} P(W_i = w_i | W_k = w_k) \\ & \quad = P(W_k = w_k) * \prod_{|k-i|=1} P(W_i = w_i | W_k = w_k) * \prod_{|k-i|>1} P_{far}(W = w_i | W_k = w_k) \end{aligned}$$

where the probabilities for $|k - i| = 1$ are from the Spaced-Bigram model (simple bigrams in this case), and the probabilities for $|k - i| > 1$ are computed like they were in the Bag-of-Bigrams model:

$$P_{far}(W = w_i | W_k = w_k) = \frac{P_{far}(W = w_i, W_k = w_k)}{P(W_k = w_k)} = \frac{\frac{1}{K-3} \sum_{|k-j|>1} c(w_i, w_k, k-j)}{c(w_k)}$$

but a normalization of $K - 3$ is used instead of $K - 1$ since the bag of words doesn't include the 3 words: $w_{k-1} w_k w_{k+1}$.

This model has a parameter space only 2 times that of normal bigrams (internally, there will be 2 tables: one for normal adjacent bigram counts, and another for "all bigrams further than 1 away" counts).

Note that this model could be generalized to transition from Spaced-Bigrams to Bag-of-Bigrams at a distance other than 1 (perhaps 2 or 3).

## 5 Experiment Process

For this project, we used the North American News Corpus - newspaper articles from the New York Times, Wall Street Journal, and others during 1994-1996. All non-alphabetic characters were converted to spaces, all letters were forced to lower-case, and all articles were simply concatenated together.

A training set of 25 million words was used to gather counts $c(w_1, w_2, d)$ for $0 \leq d \leq 4$. Recall that $c(w_1, w_2, d) = c(w_2, w_1, -d)$, so effectively we use a window of size $K = 9$ ( $-4 \leq d \leq 4$ ). The focus word was set to be the center word ($k = 5$). The training set contained roughly 140,000 unique words (token types). Surprisingly, the news corpus contains a large number of typos and misspellings. However, since they occur with low frequency, the language models never select them. Nevertheless, it is important to note that nowhere in this experiment was a dictionary used - the "dictionary" was simply all 140,000 words found in the training set.

All counts were smoothed using Laplace Smoothing (1 is added to every count, and the effective corpus size is increased appropriately). In retrospect, I noticed the counts $c(w_1, w_2, d)$ for $d > 2$ were mostly all 1's before smoothing, perhaps indicating that 25 million words is not enough training (there was more available, but time and space constraints prevented using it).

The next step was to create a test set of word windows with an incorrect spelling as the center word. A set of 100,000 9-word windows were taken from an unused portion of the news corpus. (Each window was taken 100 words after the previous window so as to "spread out" in the test cases.)

The center word of each window was "mangled" by introducing 1 random character edit. That is, a random location in the word was chosen, and an operation from

- switch the character with the one next to it
- delete the character
- insert a random character 'a' - 'z'
- change the character to a random character 'a' - 'z'

was chosen at random. This mangled focus word, along with the other original words in the window produces a "misspelling in context." The original unmangled focus word is the correct correction.

This assumes that the original focus word was correctly spelled to begin with. As we mentioned before, the news corpus does have a fair amount of typos. However, the percentage is low. Also, such "unfair" test windows should be equally difficult to all language models, so the scores of each model should still be useful for comparing them.

This also assumes (partially) that the mangled word is in fact a misspelling. It is possible that a dictionary word is mangled into another dictionary word (just as humans can make this mistake). For example, "hear" may be mistyped as "near". While traditional spell checkers would not be able to correct this, a system that uses context might.

Next, the set of words with an edit distance of exactly 1 away from a given mangled word was computed. Notice this candidate set contains the original correct word. If a non-edit occurred - such as transposing the last two letters of `‘‘three’’` - the test case was discarded. Also if the candidate set only contained one word, this trivial test case was also discarded.

Each candidate word was placed as the center word, and the likelihood $P(\widetilde{S}|S) \cdot P(S)$ was computed. $P(S)$ is where the language model is applied. $P(\widetilde{S}|S)$ is the error model. A simplifying assumption made for this experiment is that the error probability of all words the same edit distance away is the same. While not an unreasonable assumption, it is clearly false. For example, a person who typed `‘‘jero’’` is more likely to have meant `‘‘hero’’` than `‘‘zero’’` (since $h \rightarrow j$ is a likely typo, while $z \rightarrow j$ is not), even though both `‘‘hero’’` and `‘‘zero’’` are an edit distance of 1 away from `‘‘jero’’`. Thus the error model is a constant and drops out from this experiment, and the candidate that has the highest probability according to the language model is chosen. (Ties are broken randomly.)

Here is an example:

- the original window was
  `has been around for --greater-- than years homeopathy which`
- `greater` was mangled into `geater` via 1 random edit.
- the set of words exactly 1 edit away from `geater` is
  `greater heater beater eater seater getter neater grater gefter gater geter gealer weater glater teater`
- $P($ `has been around for --greater-- than years homeopathy which` $)$ is computed
- $P($ `has been around for --heater-- than years homeopathy which` $)$ is computed
- $P($ `has been around for --beater-- than years homeopathy which` $)$ is computed
- $P($ `has been around for --??-- than years homeopathy which` $)$ is computed for all candidates
- the candidate with the highest probability is selected as the replacement

Another, less photogenic example for a smaller word is:

- the original window was: `and sheet metal about --an-- hour s ride on`
- `an` was mangled into `san` via 1 random edit.

- the set of words exactly 1 edit away from `san` is
  `an can say man saw son sen sun ban ran jan van dan sam sat sa fan pan kan sad sank sand sang span sean ian sao stan tan sin han scan sani swan sap sas sane sans shan nan wan sag yan sal gan sax sant lan sac sdn sna sana sai sar sano syn sann saa osan saf sain sanh sanz zan sawn sav stn sahn sian sak suan sn slan sae scn sab sah sau tsan spn usan smn asan ssn saun uan snan sln saz sayn aan isan svan xan sany sanr sarn ksan saln saan sman saj sbn skn sgn shn`
  (Notice there is a large number of typos, acronyms, abbreviations, and other non-dictionary words.)

The above procedure was done on the same 100,000 windows for each language model. Here are the results:

| Language Model Used | Accuracy |
|---|---|
| Null | 12.9 % |
| Unigram | 74.4 % |
| Before-Bigram | 83.7 % |
| After-Bigram | 84.6 % |
| Spaced-Bigrams | 86.6 % |
| Bag Of Bigrams | 79.2 % |
| Hybrid Bigrams | 85.5 % |

Figure 2: Spelling correction accuracies on words that had a 1-edit mistake.
Language models were given 4 words of context before and after the misspelled word.

The following observations can be made:

- The majority of cases could be solved by simply picking the candidate that was most frequent in the training set. That is, the unigram model (which doesn't use any context) captured most of the statistics needed to do spelling correction.
- The simple bigram models (Before-Bigram and After-Bigram) performed noticeably better. This confirms that there is syntactic information to help in spelling correction.
- The Spaced-Bigrams model performed marginally better. It is difficult to guess whether context further than 1 word away only contains a small amount of useful information for spelling correction, or if the training corpus was not large enough and this method didn't live up to its full potential.
- The Bag-Of-Bigrams model performed *worse* than the simple bigram models despite using *more* context. The explanation for this is that syntactic information from the previous/next word is important, and the Bag-Of-Bigrams model discards the distance of the context words - throwing them all together in the same "bag."
- The Hybrid-Bigrams model (a hybrid between Spaced-Bigrams and Bag-Of-Bigrams) performed slightly worse than Spaced-Bigrams. Since it retained both the previous and next context words, it was able to outperform the simple bigram models, but it seems "hindered" by the Bag Of Words assumption made for all other words.

## 6 Conclusions

The author initially thought that spelling correction using a language model alone would be difficult, and that a large context window would be needed to accurately rank candidate replacements. Surprisingly, the task turned out to be too easy! A simple 2-word bigram performed almost as well as the best sophisticated model.

However, as the experiment stands, many of the test cases were for simple verbs and function words (since they are naturally frequent in the test corpus). One might surmise that in these cases, context outside of 1 word may not be useful. For topic-specific nouns, the more sophisticated models may perform much better than simple bigram models.

The contributions of this paper are the following:

- showing that an explicit dictionary is not needed for spelling correction. Instead, a large and possibly noisy set of words taken (unfiltered) from a training corpus works. Indeed, this is the approach taken by spelling correction systems for search engines like Google.
- showing that a simple bigram is powerful enough to correct 84% of test cases in environments similar to this experiment, without the use of an error model
- describing three bigram-based models that can use an arbitrary context window size. Although they performed only marginally better in this experiment, other situations (like the correction of more topic-specific words) may pronounce the models' strengths.

## 7 Future Work

Now that bigrams have shown to be a powerful language model for spelling correction, the next step is to design a good error model to pair it with. Continuing with the theme of acquiring statistics automatically from training corpora, the error model should do so as well. The end goal is to have a spelling correction system that can be trained in any domain or language.

## References


[BJM83] L.R. Bahl, F. Jelinek, and R.L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Journal of Pattern Analysis and Machine Intelligence*, 1983.

[BM00] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction, 2000.

[BP98] A. Berger and H. Printz. Recognition performance of a large-scale dependency-grammar language model, 1998.

[Bud00] A. Budanitsky. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures, 2000.

[CR97] Philip Clarkson and Ronald Rosenfeld. Statistical language modeling using the CMU-cambridge toolkit. In *Proc. Eurospeech '97*, pages 2707–2710, Rhodes, Greece, 1997.

[Dam64] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176, 1964.

[DLP99] Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.

[HAHR93] Xuedong Huang, Fileno Alleva, Mei-Yuh Hwang, and Ronald Rosenfeld. An overview of the sphinx-ii speech recognition system. In *ARPA Human Language Technology Workshop*, pages 81–86. Morgan Kaufmann, March 1993. published as Human Language Technology.

[Jel89] Fred Jelinek. Self-organized language modeling for speech recognition. *Readings in Speech Recognition*, 1989.

[Jel97] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[JMBB77] Fred Jelinek, Robert L. Mercer, Lalit R. Bahl, and James K. Baker. A measure of difficulty of speech recognition tasks. Technical report, 94th Meeting of the Acoustic Society of America, 1977.

[Kat87] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35:400–401, March 1987.

[KCG90] M.D. Kernighan, K.W. Church, and W.A. Gale. A spelling correction program based on a noisy channel model. In *In Proceedings of the Thirteenth International Conference on Computational Linguistics*, pages 205–210, 1990.

[Kne96] Reinhard Kneser. Statistical language modeling using avariable context length. In *Proc. Int. Conf. Spoken Language Processing*, pages 494–497, 1996.

[LRR93] R. Lau, R. Rosenfeld, and S. Roukos. Trigger-based language models: A maximum entropy approach. In *Proc. ICASSP*, volume Vol II, pages 45–48, April 1993.

[MDM91] E. Mays, F. Damerau, and R.L. Mercer. Context based spelling correction. In *Information Processing And Management*, volume 27(5), pages 517–522, 1991.

[NW96a] T. Niesler and P. Woodland. Combination of word-based and category-based language models. In *Proc. ICSLP '96*, volume 1, pages 220–223, Philadelphia, PA, 1996.

[NW96b] T. Niesler and P. Woodland. A variable-length category-based n-gram language model. In *Proc. ICASSP '96*, pages 164–167, Atlanta, GA, 1996.

[NWW98] T. Niesler, E. Whittaker, and P. Woodland. Comparison of part-of-speech and automatically derived category-based language models for speech recognition, 1998.

[Pet80] James L. Peterson. Computer programs for detecting and correcting spelling errors. *Commun. ACM*, 23(12):676–687, 1980.

[Pet86] James L. Peterson. A note on undetected typing errors. *Commun. ACM*, 29(7):633–637, 1986.

[PPMR92] Stephen Della Pietra, Vincent Della Pietra, Robert Mercer, and Salim Roukos. Adaptive language modeling using minimum discriminant estimation. In *International Conference on Acoustics, Speech and Signal Processing*, pages 633–636, San Francisco, March 1992. Also published in Proceedings of the DARPA Workshop on Speech and Natural Language, Morgan Kaufmann, pages 103106, February 1992.

[Pri90] Patti Price. Evaluation of spoken language systems: the atis domain. In *Proceedings of the third DARPA Speech and Natural Language Workshop*, 1990.

[PZ84] Joseph J. Pollock and Antonio Zamora. Automatic spelling correction in scientific and scholarly text. *Commun. ACM*, 27(4):358–368, 1984.

[Ros94] R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1994.

[Ros96] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling, 1996.

[Ros00] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here, 2000.

[Sha51] C. Shannon. Prediction and entropy of printed english. Technical Report 30, Bell Systems, 1951.

[SO00] M. H. Siu and M. Ostendorf. Variable n-grams and extensions for conversational speech language modeling. *IEEE Transactions on Speech and Audio Processing*, 8(1):63–75, 2000.

[Tou02] Kristina; Moore Robert. Toutanova. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 144–151, 2002.

[UPE] UPENN. Linguistic data consortium.

[vSW94] E. volume, Suhm, and B. Waibel. Toward better language models for spontaneous speech, 1994.