Poor Estimates of Context are Worse than None [1]

William A. Gale
Kenneth W. Church

AT&T Bell Laboratories
Murray Hill, N.J. 07974

Abstract

It is difficult to estimate the probability of a word's context because of sparse data problems. If appropriate care is taken, we find that it is possible to make useful estimates of contextual probabilities that improve performance in a spelling correction application. In contrast, less careful estimates are found to be useless. Specifically, we will show that the Good-Turing method makes the use of contextual information practical for a spelling corrector, while attempts to use the maximum likelihood estimator (MLE) or expected likelihood estimator (ELE) fail. Spelling correction was selected as an application domain because it is analogous to many important recognition applications based on a noisy channel model (such as speech recognition), though somewhat simpler and therefore possibly more amenable to detailed statistical analysis.

Background

Statistical language models were quite popular in the 1950s, but interest faded rather suddenly when Chomsky [1] argued quite successfully that statistics should not play a role in his competence model. With the recent availability of large text corpora (of 100 million words or more), there has been a resurgence of interest in empirical methods, especially in recognition applications such as speech recognition (e.g., [2]), but also in many other areas of natural language research including machine translation ([3]). The sheer size of the available corpus data is largely responsible for the revival of these techniques. Nevertheless, there is never enough data, and consequently, it is important to study the statistical estimation issues very carefully. Specifically, we will show that the Good-Turing (GT) method[4] for estimating bigram probabilities makes the use of contextual information practical in our spelling corrector application, while attempts to use the maximum likelihood estimator (MLE) or expected likelihood estimator (ELE) fail.

Previous work [5] led to the spelling correction program, *correct*. In the course of that work, we observed that human judges were reluctant to decide between alternative candidate corrections given only as much information as was available to the program, the typo and the candidate corrections. We also observed that the judges felt much more confident when they could see a line or two of context around the typo. This suggests that there is considerable information in the context.

However, it is difficult to measure contextual probabilities. Suppose, for example, that we consider just the previous word, l. (A more adequate model would need to look at considerably more than just the previous word, but even this radically over-simplified model illustrates the problem.) Then we need to measure the conditional probabilities, $Pr(l|w)$, for all w and l in the vocabulary V. The problem is that V^2 is generally much larger than the size of the corpus, N. V is at least 10^5, so V^2 is at least 10^{10}. The largest currently available corpora are about 10^8 (100 million words). Thus, we have at least 100 times more parameters than data. In fact, the problem is much worse because the data are not uniformly distributed. [2]

Correct is reviewed in the next section, as it provides the framework within which this study is done. The section also discusses estimation techniques, rules for combining multiple sources of evidence, and evaluation procedures.

Correct

The *correct* program takes a list of misspelled words (typos) as input (as might be produced by the Unix® *spell* program), and outputs a set of candidate corrections for each typo, *along with* a probability. These probability scores distinguish *correct* from other spelling correction programs, that output a (long) list of candidiate

[1] We would like to acknowledge Mark Kernighan's work on *correct*, which laid the groundwork for this study of context modelling. We thank Jill Burstein for help with the judging.

[2] One might think that the sparse data problem could be solved by collecting larger corpora, but ironically, the problem only gets worse as we look at more data. The vocabulary is not fixed; both N and V grow as we look at more data. The rate of growth is still a matter of debate, but the evidence clearly shows that $V > O(\sqrt{N})$, and therefore, the sparse data problems only get worse as we look at more and more data.

corrections, many of which are often extremely implausible.

Here is some sample output:

Typo	Corrections
laywer	lawyer (100%) layer (0%) lawer (0%)
negotations	negotiations
notcampaigning	???
progession	progression (94%) procession (4%) profession (2%)

The entry *???* indicates that no correction was found.

The first stage of *correct* finds candidate corrections, c, that differ from the typo t by a single insertion, deletion, substitution or reversal. For example, given the input typo, *acress*, the first stage generates candidate corrections in the table below. Thus, the correction *actress* could be transformed by the noisy channel into the typo *acress* by replacing the t with nothing, @, at position 2. (The symbols @ and # represent nulls in the typo and correction, respectively. The transformations are named from the point of view of the correction, not the typo.) This unusually difficult example was selected to illustrate the four transformations; most typos have just a few possible corrections, and there is rarely more than one plausible correction.

Typo	Correction	Transformation			
acress	actress	@	t	2	deletion
acress	cress	a	#	0	insertion
acress	caress	ac	ca	0	reversal
acress	access	r	c	2	substitution
acress	across	e	o	3	substitution
acress	acres	s	#	4	insertion
acress	acres	s	#	5	insertion

Each candidate correction is scored by the Bayesian combination rule $Pr(c)\ Pr(t|c)$, and then normalized by the sum of the scores for all proposed candidates. Care must be taken in estimating the prior because of sparse data problems. It is possible (and even likely) that a proposed correction might not have appeared in the training set. Some methods of estimating the prior would produce undesirable results in this case. For example, the maximum likelihood estimate (MLE) would estimate $Pr(c) = 0$, and consequately, many candidate corrections would be rejected just because they did not happen to appear in the training set. We will encounter even more severe forms of the sparse data problem when we consider context.

We will consider four estimation methods for dealing with the sparse data problems. All of these methods attempt to estimate a set of probabilities, p, from observed frequencies, r. It is assumed that the observed frequencies are generated by a binomial process with N total observations. The estimation methods generate an adjusted frequency r^*, where r^* is a function of r. Once r^* has been determined, then p is estimated as $p \approx r^*/N^*$. $N^* = \sum r^*\ N_r$ where N_r is the frequency of frequency r, assuring that the estimated probabilities add to one. The maximum likelihood estimator (MLE) sets $r^* = r$. The MLE estimate is particularly poor when $r = 0$, since the true probabilities are almost certainly greater than 0.

Following Box and Tiao [6] , we can assume an uninformative prior and reach a posterior distribution for p. Using the expectation of this distribution amounts to using $r^* = r + .5$. We call this the expected likelihood estimate (ELE). This method is often used in practice because it is easy to implement, though it does have some serious weaknesses. The third method is the minimax (MM) method [7] , which sets $r^* = r + .5\sqrt{N}$. Its derivation is based on a risk analysis; it minimizes the maximum quadratic loss. The fourth method is the Good-Turing (GT) method [4] , which sets $r^* = (r+1)\ N_{r+1}/N_r$. Unlike the MLE, all three other methods assign nonzero probabilities, even when $r = 0$. This is probably a desirable property.

We use the ELE for the probabilities of single words as they are frequent enough not to require elaborate treatment. The channel probabilities, $Pr(t|c)$, are computed from four confusion matrices: (1) $del[x,y]$, the number of times that the characters xy (in the correct word) were typed as x in the training set, (2), $add[x,y]$, the number of times that x was typed as xy, (3) $sub[x,y]$, the number of times that y was typed as x, and (4) $rev[x,y]$, the number of times that xy was typed as yx. Probabilities are estimated from these matrices by using $chars[x,y]$ and $chars[x]$, the number of times that xy and x appeared in the training set, respectively, as the total number of observations appropriate to some cell of a matrix. The probabilities are estimated using the Good-Turing method [4] , with the cells of the matrices as the types.

Returning to the *acress* example, the seven proposed transformations are scored by multipling the prior probability (which is proportial to 0.5 + column 4 in the table below) and the channel probability (column 5) to form a raw score (column 3), which are normalized to produce probabilities (column 2). The final results is: *acres* (45%), *actress* (37%), *across* (18%), *access* (0%), *caress* (0%), *cress* (0%). This example is very hard; in fact, the second choice is probably right, as can be seen from the context: *...was called a "stellar and versatile* **acress** *whose combination of sass and glamour has defined her....* The program would need a much better prior model in order to handle this case. The next section shows how the context can be used to take advantage of the fact that that *actress* is considerably more plausible than *acres* as an antecedent for *whose.*

c	%	Raw	freq(c)	Pr(t|c)
actress	37%	.157	1343	55./470,000
cress	0%	.000	0	46./32,000,000
caress	0%	.000	4	.95/580,000
access	0%	.000	2280	.98/4,700,000
across	18%	.077	8436	93./10,000,000
acres	21%	.092	2879	417./13,000,000
acres	23%	.098	2879	205./6,000,000

Many typos such as *absorbant* have just one candidate correction, but others such as *adusted* are more difficult and have multiple corrections. (For the purposes of this experiment, a typo is defined to be a lowercase word rejected by the Unix® spell program.) The table below shows examples of typos with candidate corrections sorted by their scores. The second column shows the number of typos in a seven month sample of the AP newswire, broken out by the number of candidate corrections. For example, there were 1562 typos with exactly two corrections proposed by *correct.* Most typos have relatively few candidate corrections. There is a general trend for fewer choices, though the 0-choice case is special.

#	Freq	Typo	Corrections
0	3937	admininistration	
1	6993	absorbant	absorbent
2	1562	adusted	adjusted dusted
3	639	ambitios	ambitious ambitions ambition
4	367	compatability	compatibility compactability comparability computability
5	221	afte	after fate aft ate ante
6	157	dialy	daily diary dials dial dimly dilly
7	94	poice	police price voice poise pice ponce poire
8	82	piots	pilots pivots riots plots pits pots pints pious
9	77	spash	splash smash slash spasm stash swash sash pash spas

We decided to look at the 2-candidate case in more detail in order to test how often the top scoring candidate agreed with a panel of three judges. The judges were given 564 triples (e.g., *absurb, absorb, absurd*) and a concordance line (e.g., *...financial community. "It is* **absurb** and probably obscene for any person so engaged ,to...). The first word of the triple was a *spell* reject, followed by two candidates in alphabetical order. The judges were given a 5-way forced choice. They could circle any one of the three words, if they thought that was what the author had intended. In addition, they could say "other" if they thought that some other word was intended, or "?" if they were not sure what was intended. We decided to consider only those cases where at least two judges circled one of the two candidate corrections, and they agreed with each other. This left only 329 triples, mainly because the the judges often circled the first word, indicating that they thought it had been incorrectly rejected by *spell.*

The following table shows that *correct* agrees with the majority of the judges in 87% of the 329 cases of interest. In order to help calibrate this result, three inferior methods are also evaluated. The *channel-only* method ignores the prior probability. The *prior-only* method ignores the channel probability. Finally, the *neither* method ignores both probabilities and selects the first candidate in all cases. As the following table shows, *correct* is significantly better than the three alternative methods. The table also evaluates the three judges. Judges were only scored on triples for which they selected one of the proposed alternatives, and for which the other two judges agreed on one of the proposed alternatives. A triple was scored "correct" for one judge if that judge agreed with the other two and "incorrect" if that judge disagreed with the other two. The table shows that the judges significantly out-perform *correct,* indicating that there is room for improvement.

Method	Discrimination	%
correct	286/329	87 ± 1.9
channel-only	263/329	80 ± 2.2
prior-only	247/329	75 ± 2.4
chance	172/329	52 ± 2.8
Judge 1	271/273	99 ± 0.5
Judge 2	271/275	99 ± 0.7
Judge 3	271/281	96 ± 1.1

Context

As previously noted, the judges were extremely reluctant to cast a vote without more information than *correct* uses, and they were much more comfortable when they could see a concordance line or two. This suggests that contextual clues might help improve performance. However, it is important to estimate the context carefully; we have found that poor measures of context are worse than none.

In this work, we use a simple n-gram model of context, based on just the word to the left of the typo, l, and the word to the right of the typo, r. Although n-gram methods are much too simple (compared with much more sophisticated methods used in AI and natural language processing), even these simple methods illustrate the problem that poor estimates of contextual probabilities are worse than none. The same estimation issues are probably even more critical when the simple n-gram models of context are replaced by more sophisticated AI models.

The variables l and r are introduced into the Baysian scoring function by changing the formula from $Pr(c)Pr(t|c)$ to $Pr(c)Pr(t,l,r|c)$, which can be approximated as $Pr(c)Pr(t|c)Pr(l|c)Pr(r|c)$, under appropriate independence assumptions. The issue, then, is how to estimate the two new factors: $Pr(l|c)$ and $Pr(r|c)$. We have four proposals: MLE, ELE, MM and GT. Let us consider one way of using the ELE method first. It is straightforward and similar to our best method, but hopelessly wrong.

$$\begin{aligned} Pr(l|c) &= \frac{Pr(lc)}{Pr(c)} \\ &\approx \frac{(freq(lc)+0.5)/d_1}{(freq(c)+0.5)/d_2} \\ &\propto \frac{freq(lc)+0.5}{freq(c)+0.5} \end{aligned}$$

where $d_1 = N+V^2/2$ and $d_2 = N+V/2$. We can ignore the constant d_2/d_1 and use the proportion to score candidate corrections. Similarly, we use the relation $Pr(r|c) \propto (freq(cr)+0.5)/(freq(c)+0.5)$ for the right context. When these estimates for $Pr(l|c)$ and $Pr(r|c)$ are substituted in the formula, $Pr(c)Pr(t|c)Pr(l|c)Pr(r|c)$, we have:

$$\frac{Pr(t|c)\ (freq(lc)+0.5)\ (freq(cr)+0.5)}{(freq(c)+0.5)} \qquad \text{E/E}$$

This new formula produces the desired results for the *acress* example, as illustrated in the following table. (The column labeled raw is 10^6 times the formula E/E, as only proportionalities matter.) Note that *actress* is now prefered over *acres* mostly because *actress whose* is more common than *acres whose* (8 to 0). Presumably the difference in frequencies reflects the fact that *actress* is a better antecedent of *whose*. Note also though, that *cress* is now considered a plausible rival because of errors introduced by the ELE method. The high score of *cress* is due to the fact that it was not observed in the corpus, and therefore the ELE estimates $Pr(l|c) = Pr(r|c) = 1$, which is clearly biased high.

c	%	Raw	freq(c)	Pr(t|c)	freq(lc)	freq(cr)
actress	69%	1.85	1343	55./470,000	2	8
cress	27%	.719	0	46./32,000,000	0	0
caress	3%	.091	4	.95/580,000	0	0
access	0%	.000	2280	.98/4,700,000	2	0
across	0%	.011	8436	93./10,000,000	0	20
acres	0%	.003	2879	417./13,000,000	0	0
acres	0%	.003	2879	205./6,000,000	0	0

We will consider five methods for estimating $Pr(l|c)$. The method just described is called the E/E method, because both $Pr(lc)$ and $Pr(c)$ are estimated with the ELE method. The M/E method uses the MLE estimate for $Pr(lc)$ and the ELE estimate for $Pr(c)$. The E method takes $Pr(lc)$ proportional to the ELE estimate $(freq(lc)+0.5)$, but the denominator is adjusted so that $\sum_c Pr(l|c) = 1$. The MM method adjusts the minimax suggestion in [7] in the same way. The G/E method uses the enhanced Good-Turing (GT) method for $Pr(lc)$ and the ELE estimate for $Pr(c)$.

$$Pr(l|c) = \frac{Pr(lc)}{P(c)} \approx \frac{freq(lc)+0.5}{freq(c)+0.5} \qquad \text{E/E}$$

$$Pr(l|c) = \frac{Pr(lc)}{P(c)} \approx \frac{freq(lc)}{freq(c)+0.5} \qquad \text{M/E}$$

$$Pr(l|c) \approx \frac{freq(lc)+0.5}{freq(c)+V/2} \qquad \text{E}$$

$$Pr(l|c) \approx \frac{freq(lc)+0.5\sqrt{freq(c)}}{freq(c)+0.5V\sqrt{freq(c)}} \qquad \text{MM}$$

$$Pr(l|c) \approx \frac{(r+1)\ \dfrac{N_{r+1}}{N_r}}{freq(c)+0.5} \qquad \text{G/E}$$

The first two methods are useless, as shown by the performance of the context alone:

Poor Estimates of Context Offer Little or No Help

	chance	M/E	E/E
wrong	164.5	15	169
uninformative	0	136	4
right	164.5	178	156

The other three are better. The performance of G/E is significantly better than the other four.

Better Estimates of Context Exist

	E	MM	G/E
wrong	62	59	45
uninformative	0	0	4
right	267	270	280

For the Good-Turing estimates, we use an enhanced version of the Good-Turing estimator. The basic estimator is applied to subgroups of the bigrams. The subgroups have similar values of Np_xp_y, where p_x and p_y are the probabilities for the individual words. The grouping variable is the expected frequency of the bigram if the words occurred independently. Its use is discussed in detail by [8] It results in about 1400 significantly different estimates for bigrams not seen in the training text, and in about 150 different estimates for words seen once.

When combined with the prior and channel, G/E is the only one of the five estimation methods that improves significantly[3] on the performance of *correct*. The following table shows *correct* in column 1, followed by the two disastrous measures M/E and E/E, then the two useless measures E and MM, and finally the one useful measure G/E.

Context is Useless Unless Carefully Measured

		disastrous		useless		useful
	no	+M/E	+E/E	+E	+MM	+G/E
	context	context	context	context	context	context
wrong	43	11	61	39	40	34
useless	0	136	0	0	0	0
right	286	182	268	290	289	295
%	86.9	55.3	81.5	88.1	87.8	89.7
± σ	1.9	2.7	2.1	1.8	1.8	1.7

Conclusions

We have studied the problem of incorporating context into a spelling correction program, and found that the estimation issues need to be addressed very carefully. Poor estimates of context are useless. It is better to ignore context than to model it badly. Fortunately, there are good methods such as G/E that provide a significant improvement in performance. However, even the G/E method does not achieve human performance, indicating that there is considerable room for improvement. One way to improve performance might be to add more interesting sources of knowledge than simple n-gram models, e.g., semantic networks, thesaurus relations, morphological decomposition, parse trees. Alternatively, one might try more sophisticated statistical approaches. For example, we have only considered the simplest Baysian combination rules. One might try to fit a log linear model, as one of many possibilities. In short, it should be taken as a challenge to researchers in computational linguistics and statistics to find ways to improve performance to be more competitive with human judges.

References

1. Chomsky, N., *Syntactic Structures,* Mouton & Co, The Hague (1957).
2. Nadas, A., ''Estimation of probabilities in the language model of the IBM speech recognition system,'' *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-32** pp. 859-861 (1984).
3. Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., and Pietra, P., ''A Statistical Approach to French/English Translation,'' in *Proceedings RIAO88 Conference on User-oriented Content-based Text and Image Handling,* RIAO, Cambridge, Massachusetts (March 21-24, 1988).
4. Good, I. J., ''The population frequencies of species and the estimation of population parameters,'' *Biometrika* **40** pp. 237-264 (1953).
5. Kernighan, M. D, Church, K. W., and Gale, W. A., ''A Spelling Corrector Based on Error Frequencies,'' in *Proceedings of the Thirteenth International Conference on Computational Linguistics,* (1990).
6. Box, G. E. P. and Tiao, G. C., *Bayesian Inference in Statistical Analysis,* Addison-Wesley, Reading, Massachusetts (1973).
7. Steinhaus, H., ''The problem of estimation,'' *Annals of Mathematical Statistics* **28** pp. 633-648 (1957).
8. Church, K. W. and Gale, W. A., ''Enhanced Good-Turing and Cat-Cal: Two New Methods for Estimating Probabilities of English Bigrams,'' *Computer, Speech, and Language,* (1991).

[3] The GT method changes the program's preference in 25 of the 329 cases; 17 of the changes are right and 8 of them are wrong. The probability of 17 or more right out of 25, assuming equal probability of two alternatives, is .04. Thus, we conclude that the improvement is significant.