What's Wrong with Adding One?

William A. Gale
Kenneth W. Church
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ, 07974

1. Introduction

One could estimate the probability of a word (or n-gram) in English by collecting a large corpus of English text, counting the number of times the word appears in the corpus, and normalizing by the size of the corpus. Unfortunately this method (known as MLE) produces a poor estimate when the count is small and an unacceptable estimate when the count is zero. This is particularly problematic in many applications because counts are often small and zero is often the most frequent count.

A common engineering practice, first proposed by Jeffries (1948), adds one to the counts in order to "fix" the zeros. This method, though, has severe deficiencies as will be illustrated by example: the estimation of word bigram probabilities in a year of Associated Press newswire (N = 44 million words). The source of these deficiencies can be shown by comparing the Add-One method to the Good-Turing method. This comparison reveals that adding one is correct only if the ratio of unseen to observed types equals the ratio of all types to sample size. Since these ratios will be equal only by coincidence, adding one will obtain the correct answers only by happenstance, if at all.

2. Problem: The Expected Frequency of Unseen Types

This chapter deals with a problem occuring in corpus linguistics. Many linguistic applications use a prior model of language to disambiguate otherwise equally probable outputs. These applications include corpus tagging (Leech, this volume), word sense disambiguation (Gale, Church and Yarowsky, 1993), spelling correction (Church and Gale, 1991b), machine translation (Brown *et al.*), speech recognition (Katz, 1987; Nàdas, 1984), and others. It is common practice to use tables of probabilities of single words, pairs of words, and triples of words (*n-grams*) as a prior model. The problem these models face, however, is to estimate probabilities for n-grams which were not observed in the training corpus. For there will always be such n-grams, however large the corpus.

2.1 Zero, the maximum likelihood estimator (MLE), is unacceptable

In principle, n-gram probabilities can be estimated from a large sample of text, by counting r, the number of occurrences of each n-gram of interest, and dividing by N, the size of the training sample. This method, which is known as the ''maximum likelihood estimator'' (MLE) is very simple. However, it is unacceptable because n-grams which do not occur in the training text are assigned zero probability. This is qualitatively wrong for use as a prior model, because it would never allow the n-gram, while clearly some of the unseen n-grams will occur in other texts. For non-zero frequencies, the MLE is quantitatively wrong.

We consider methods that tackle this problem by adjusting the observed frequency. Let r^* be the adjusted frequency for a type observed r times. Then p, the probability of the type, is estimated by r^*/N. In order to satisfy the constraint $\Sigma p = 1$, the adjusted frequencies must satisfy

$$\frac{\Sigma r^*}{N} = 1. \qquad 1.1$$

2.2 Add-One as a potential solution

Johnson (1932) and Jeffreys (1948) proposed statistically motivated approaches to estimation of r^*. Johnson suggested adding some constant k to the frequency for each type and renormalizing appropriately. That is, the adjusted frequency, r^*, is $r+k$ times a renormalization factor, $\frac{N}{N+kS}$, where S is the total number of types. In this chapter we consider the special case proposed by Jeffries, where $k = 1$. That is, the *Add-One* method is defined by:

$$r^* = (r+1)\,\frac{N}{N+S} \qquad 1.2$$

This is also a common engineering practice, although there is little discussion in the literature, and its use is often not mentioned. The work by Church (1989) used it at one point, although the paper did not describe the use.

3. The Example: Estimating Bigram Probabilities in the 1988 AP Newswire

We have recently studied the estimation of probabilities for English bigrams (Church and Gale, 1991a). This task is useful for testing methods such as Add-One.

Our corpus was selected from articles distributed by the Associated Press newswire in 1988, which we refer to as the ‘‘AP wire.’’ Some portions of the year were lost. The remainder was processed automatically (Liberman and Riley, 1988) to remove identical or nearly identical articles. There remained N = 44 million words in the corpus.

We split the 1988 AP wire into two portions, one for estimating probabilities and one for testing. We made the split by taking bigrams beginning with even numbered words as one sample, those beginning with odd numbered words as the other. It is important that we made this split by taking every other bigram, because the topics discussed in the AP wire generate measurable differences over the period of a month. By taking every other bigram, we generate as close to two samples of the same universe of discourse as possible.

When we speak of ‘‘words,’’ we use a common term to hide a number of processing decisions. Roughly, a word is a string of characters delimited by white space. For instance, ‘‘The’’ and ‘‘the’’ are different words, and ‘‘need’’ and ‘‘needs’’ are also. Punctuation modifies this definition: period, comma, hyphen and other punctuation marks were treated as words. Additional tokens were inserted automatically to delimit sentences, paragraphs and discourses. These definitions resulted in a vocabulary size, V, for 1988 of 400,653 words. The resulting vocabulary size is two orders of magnitude larger than the 5000 words reported for the early IBM speech recognizer (Nàdas, 1984).

We regard the model for unigrams as completely fixed before beginning to study bigrams. This includes specifying the vocabulary, $\mathbf{V}$, its cardinality, V, and an estimate, $e(p(x))$, of the probability, $p(x)$, of each word, x, in $\mathbf{V}$. We also suppose that the variances of the estimates in the unigram model are known. Likewise, we would regard a bigram model as fixed before studying a trigram model, if we were to study such a model.

We obtain a number of substantially different cases by dividing the set of V^2 bigrams by values of the variable $jii \equiv Ne(p(x))\, e(p(y))$ defined for each bigram *xy* from the unigram model. *Jii* is an acronym for ‘‘joint if independent’’; it is the expected frequency in a corpus of size *N* if each word of the corpus were selected independently with probabilities given by the unigram model. We form 36 different cases by forming bins of words with approximately equal values of jii, taking three bins per order of magnitude.

3.1 A Standard

As mentioned above, the MLE method produces poor estimates when the counts are small and most of our bigrams have small counts. Fortunately, since there are so many bigrams with the same small count, one can study them as a group. That is, split the text into two halves, use the first half to categorize the bigrams into groups of bigrams all occurring the same number of times, and then use the second half to calibrate the probabilities of each group. In other words, each bigram *b* is assigned to a group of bigrams that occur *r* times in the first half. (That is, we select the group defined by $b | r_1(b) = r$ where $r_1(b)$ is the frequency with which bigram *b* appears in the first half.) Let N_r denote the size of the group. (Note that $N_r = \sum_{b | r_1(b) = r} 1$.) We then look at the second half and count C_r, the total occurrences of these bigrams in the second half. (That is, $C_r \equiv \sum_{b | r_1(b) = r} r_2(b)$, where the $r_2(b)$ is the observed frequency of the bigram, *b*, in the second half of the text.) The bigram *b* is then assigned an adjusted frequency r^* of C_r / N_r. This method was first described by Jelinek and Mercer (1985), who termed it the *held Out* method. We will use it as a standard for comparing the Good-Turing and Add-One methods.

3.2 Add-One fails for unseen bigrams

To highlight the inadequacy of Add-One, we will compare its performance to another method called the Good-Turing method (Good 1953):

$$r^* = (r+1)\ \frac{E(N_{r+1})}{E(N_r)} \qquad 2.1$$

where N_r is the number of bigrams that occur *r* times, and $E(x)$ is the expectation of the random variable *x*. The proof of this equation sketched by Good (1953), and made rigorously by Church, Gale, and Kruskal (1991), depends critically on the assumption that each type has an independent binomial distribution. This assumption is usually false for natural language due to clumping of words relevant to a discourse, as shown by Mosteller and Wallace (1964). The way we split our corpus into two parts, taking alternate bigrams, has made the assumption close to true. Equation 2.1 differs from the equation given by Good (1953) in making the expectation operators clear. Without them, the equation is false. However, we do not know the expectations that the formula calls for. Good (1953) spent about a quarter of the paper discussing ways to smooth the N_r, which is the best way known to estimate the expectations required by the formula.

Add-One Fails for r=0

Figure 1. This figure compares estimates by Add-One and Good-Turing methods to a standard for bigrams that did not occur in the training sample. The standard was computed by the Held Out method. Asterisks show the observed standard for each of 36 cases distinguished by jii, a frequency estimate based on the unigram model. The predictions for Add-One are shown with a solid line, while those for the Good-Turing estimator are shown with a dashed line. The Add-One predictions miss the standard by up to three orders of magnitude, while the Good-Turing estimates agree very closely with the standard.

Figure 1 shows that Add-One is a poor estimator for r=0. The Good-Turing estimates agree closely with the standard in every jii bin, but Add-One differs by factors ranging from 10 to 3000.

3.3 Add-One estimates are worse than MLE

While we gather the data for the standard, we also collect the variance of the observed frequencies in the validation sample. Church and Gale (1991a) show that Good-Turing theory predicts this variance to be

$$v_{jr}^{GT} = r^*(1 + (r+1)^* - r^*) \qquad 2.2$$

and that this prediction is accurate.

Given these variances, it is natural to compare other estimates to the standard estimates by comparing their difference to the variance: $t_{jr} = (r^* - r_{jr}^S)/\sqrt{v_{jr}^{GT}}$. For each jii bin, j, and each frequency, r, this equation defines a t-score, t_{jr}, for the difference of an estimate, r^*, from the standard estimate, r_{jr}^S. We condense this array of information into a single graph by forming the root mean square (RMS) t-score:

$$R_r = \sqrt{\frac{1}{J} \sum_{j=1}^{J} t_{jr}^2}$$

A perfect predictor would give RMS t-scores of about one, because the variance of one standard observation is used as the denominator.

Add-One is Worse than MLE

Figure 2. The figure shows RMS standardized errors for three estimation methods. A perfect predictor would have an RMS error of one, and all of the methods approach that as frequency increases. However, the Good-Turing estimates achieve the ideal performance for smaller frequencies than does the MLE. Add-One does not achieve ideal performance in the range shown.

Figure 2 shows that Add-One is a poor estimator for $r \leq 100$. The Good-Turing estimator is nearly a perfect estimator over the range of frequencies $r > 15$. Even the MLE is better than Add-One, because Add-One has added so much to the estimates for unseen bigrams that it seriously underestimates the bigrams that were seen. The figure suppresses values for MLE and Add-One that exceed the maximum RMS t-value for Good-Turing in the interests of clarity of presentation of smaller values. The ranking shown in the figure holds for all the small frequencies so suppressed.

4. Why Add-One Fails

The close match between Good-Turing estimates and the standard, as shown in Figures 1 and 2, validates the Good-Turing formula. We use the Good-Turing equation here to derive two relations required for Add-One to give accurate answers. The empirical failure of these required relations explains the poor performance of Add-One shown in Figures 1 and 2.

Equating the Add-One prediction (equation 1.2) and the Good-Turing prediction (equation 2.1) gives

$$(r+1)\frac{N}{N+S} = (r+1)\frac{N_{r+1}}{N_r}, \qquad r = 0, \cdots \qquad 3.1$$

>From this it follows that

$$\frac{N_{r+1}}{N_r} = \frac{N}{N+S}, \qquad r=0, \cdots \tag{3.2}$$

Letting $\rho = \frac{N}{N+S}$,

$$N_1 = \rho N_0$$

$$N_2 = \rho N_1 = \rho^2 N_0$$

. . .

$$N_r = \rho^r N_0 \tag{3.3}$$

That is, for Add-One to give accurate answers, the N_r must form a geometric series. The following figure shows that bigram data do not give a geometric series.

Add-One Requires Nr to be a Geometric Series

Figure 3. For the Add-One estimator to give accurate answers, it is necessary that N_r, the frequency of frequency r, be given by the geometric series $N_r = \rho^r N_0$ with $\rho = N/N+S$. For jii bin 22, this figure shows that the data differ strongly from the required relationship.

The geometric relationship fails for our bigram data. It seems likely to fail for any linguistic data since linguistic data are commonly found to obey Zipf's law. The Generic-Specific form of Zipf's law predicts that $\log N_r$ is a linear function of r, whereas the geometric series required for Add-One would be a linear plot on semi-log scales. Still, the geometric series might hold for some other domain.

The second relation required is an equality of two ratios:

$$\frac{\text{unseen types}}{\text{observed types}} = \frac{\text{total types}}{\text{training sample size}} \tag{3.4}$$

or symbolically,

$$\frac{N_0}{S - N_0} = \frac{S}{N} \tag{3.5}$$

To prove this we will assume (as is required for Add-One to be a good estimator) that $N_r = N_0 \rho^r$ where $\rho = \frac{N}{N+S}$, even though this assumption appears to fail for our particular data. Now,

$$S = \sum_{r=0}^{\infty} N_r = \sum_{r=0}^{\infty} N_0 \rho^r = N_0 \frac{1}{1-\rho} = N_0 \frac{N+S}{S} \tag{3.6}$$

So

$$\frac{S}{N} = \frac{\frac{S}{S+N}}{1 - \frac{S}{S+N}} = \frac{\frac{N_0}{S}}{1 - \frac{N_0}{S}} = \frac{N_0}{S - N_0} \tag{3.7}$$

The following figure compares these two ratios for the bigram data.

Add-One Requires Equality of Two Ratios

Figure 4. For Add-One to give accurate answers, the ratio of unseen types to observed types must equal the ratio of all types to training sample size. This figure shows that these two ratios are never equal for our 36 cases.

There is no logical relationship between the ratios shown here. An equality would simply be a coincidence in any study, although it might hold for some data sets. The failure of this relationship seems likely to be more general than linguistic data.

5. Conclusions

Add-One has severe deficiencies: it misses the mark for unseen bigrams by up to three orders of magnitude (and is never close), and has larger errors than even the maximum likelihood estimator. One reason for this poor performance is that the bigram frequency of frequencies, N_r, follow a power law (linear on log-log scales) rather than a geometric law (linear on semi-log scales).

In addition, for Add-One to produce reasonable estimates, it is necessary that the ratio of unseen types to observed types and the ratio of all types to the training sample size be equal. Since there is no reason for a relationship between sample size and the population surveyed, this condition is usually invalid.

With the availablity of accurate estimators, such as the Good-Turing estimator, there is no reason to use such a poor estimator as Add-One.

Acknowledgement

This is a revision and first publication of the previously available AT&T Bell Laboratories Statistical Research Report No. 90, November, 1989.

References

Brown, P., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jellinek, J. Lafferty, R. Mercer, and P. Roosin (1990), ‘‘A Statistical Approach to Machine Translation,’’ *Compuational Linguistics*, v. 16, pp. 79-85.

Church, K. W. (1989), ‘‘A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text,’’ in IEEE 1989 International Conference on Acoustics, Speech, and Signal Processing, MAY 23-26 1989 Glasgow, Scotland, U.K.

Church, K. W., and W. A. Gale (1991a), ‘‘A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams,’’ *Computer, Speech, and Language*, v. 5, pp. 19-54.

Church, K. W., and W. A. Gale (1991b), ‘‘Probability Scoring for Spelling Correction,’’ *Statistics and Computing*, v. 1, pp. 93-103.

Church, K. W., W. A. Gale, and J. B. Kruskal (1991), Appendix A to Church and Gale (1991a).

Gale, W. A., K. W. Church, and D. E. Yarowsky (1993), ‘‘A Method for Disambiguating Word Senses in a Large Corpus,’’ *Computers and the Humanities*, v. 26, pp. 415-439.

Good, I.J., (1953), ‘‘The population frequencies of species and the estimation of population parameters,’’ *Biometrika*, v. 40, pp. 237-264.

Jeffreys, H., (1948) *Theory of Probability,* second edition, section 3.23, Oxford: Clarendon Press.

Jelinek, F., and R. Mercer (1985) ‘‘Probability Distribution Estimation from Sparse Data,’’ *IBM Technical Disclosure Bulletin*, v. 28, 2591-2594.

Johnson, W. E., (1932) Appendix (edited by R.B. Braithwaite) to ‘‘Probability: deductive and inductive problems,’’ *Mind,* v. 41, pp. 421-423

Katz, S. M., (1987), ‘‘Estimation of probabilities from sparse data for the language model component of a speech recognizer,’’ *IEEE Transactions on Acoustics, Speech, and Signal Processing,* v. ASSP-35, pp. 400-401.

Liberman, M. Y., and M. Riley, (1988) personal communication.

Mosteller, F., and D. Wallace (1964) *Inference and Disputed Authorship: The Federalist,* Addison-Wesley, Reading, Mass. The original version is out of print, but a slightly revised version *Applied Bayesian and Classical Inference: The case of the* Federalist *papers,* published by Springer-Verlag, New York, in 1984 is in print.

Nàdas, A., (1984), ‘‘Estimation of probabilities in the language model of the IBM speech recognition system,’’ *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. ASSP-32 pp. 859-861.