Enhanced Good-Turing and Cat-Cal: Two New Methods for Estimating Probabilities of English Bigrams

Kenneth W. Church
William A. Gale

AT&T Labs - Research
Murray Hill, NJ 07974

1. Introduction

1.1 Possible applications of bigram frequencies

A planned use of bigram frequencies is for disambiguation of the output of an optical character recognizer designed by Baird (Kahan, Pavlidis, and Baird, 1987). Consider these two simple examples. In each case, suppose the recognizer has assigned about equal probability to having recognized ''farm'' and ''form,'' having used both the unigram frequencies and the optical information. In one case, the two possibilities including adjacent words are:

federal farm credit
federal form credit

In the other case, the two possibilities are:

some form of
some farm of

We doubt that the reader has any trouble specifying which alternative is more likely. By using the following probabilities for the eight bigrams in these sequences, a computer program can rely on an estimated likelihood to make the same distinction.

bigram	probability $\times 10^6$
federal farm	0.50
federal form	0.039
farm credit	0.13
form credit	0.026
some form	4.1
some farm	0.63
form of	34.
farm of	0.81

The deciding scores will then be found by multiplying the probabilities of the the two constituent bigrams. This is only exact if the bigrams are independent, but it is useful in any case. For the first example, the scores so generated are $(0.5 \times 10^{-6}) \times (0.13 \times 10^{-6}) = 0.065 \times 10^{-12}$, and $(0.039 \times 10^{-6}) \times (0.026 \times 10^{-6}) = 0.0010 \times 10^{-12}$. The ratio of these shows that the reading ''farm'' is 65 times as likely as the reading ''form'' in the first context. In the second context, the reading ''form'' is 273 times as likely as ''farm.'' This example shows how likelihood ratios of bigram probabilities derived from a corpus of English text can be used to disambiguate optically confusable minimal pairs.

A similar application is the disambiguation of confusable pairs in a speech recognition system. Other applications include refining proof reading tools, such as the Writer's Workbench (Cherry, 1981). For example, while spelling programs have proven quite useful, many spelling errors produce another word. A program with bigram probabilities could use context to detect legal words in unlikely contexts. Labeling parts of speech is useful for proof reading tools, and one method, Church (1989), relies on estimates of probabilities of trigrams of parts of speech. Sproat and Shih (1990), relies on bigrams of Chinese characters. In addition, bigram probabilities could be used to support lexicographic research, by identifying ''interesting'' pairs of words in a particular topic relative to general use (Church and Hanks, 1989). There are more applications in information retrieval; the ability to identify the pairs common within a given document but generally uncommon provides a way to locate documents similar to a given document.

Applications are discussed further elsewhere (e. g. Church, Gale, Hindle, and Hanks, 1989); the work reported here develops and tests two methods.

1.2 Possible uses for variances

This paper uses variances of bigram estimates to compare methods. Other applications are possible. For instance, suppose a lexicographer wishes to identify words used with the word ‘‘potatoes.’’ The following table shows some bigrams which end with ‘‘potatoes,’’ ranked by t-scores comparing the joint probability of the bigram to chance. Variances are required to calculate the t-scores.

word-1	word-2	*t*
sweet	potatoes	4.6
mashed	potatoes	4.3
,	potatoes	4.3
and	potatoes	4.0
couch	potatoes	3.8
of	potatoes	3.3
frozen	potatoes	3.3
fresh	potatoes	2.8
small	potatoes	2.8
baked	potatoes	2.1

It should be clear that the variances allow the lexicographer to identify many words of interest, easily. Pairs with higher scores are more likely to be interesting pairs, so the lexicographer will find more interesting pairs in a given amount of time by examining the pairs in the order of the t-scores than by using a random order.

1.3 The failure of the maximum likelihood estimator (MLE)

Suppose we are told only that a particular pair of words occurred r times in a sample of N pairs of the language. The obvious estimator of the probability of the pair is r/N. This estimator is in fact the maximum likelihood estimator if the occurrence of the pair of interest is a random variable with a binomial distribution.

The problem is that most possible pairs will not occur in a given sample, because, as shown in section 2, the square of the vocabulary size greatly exceeds the number of words in the corpus. Therefore, most of the possible pairs have an observed frequency $r = 0$. However, as our example shows, we wish to take products and ratios of the estimated probabilities, and it will often happen that neither of two alternative bigrams has been seen in the training text. In this case, the MLE would not determine a likelihood ratio, as both probabilities would be estimated at zero. Furthermore, even the same estimate for all the unseen bigrams is seriously misrepresentative, since our work shows predictable factors of a million among the probabilities of unseen bigrams.

The maximum likelihood estimates of variances are also biased, because the estimates for the probabilities are severely biased. In particular, no estimate of the variance is available for bigrams not seen in the training corpus.

1.4 Some alternatives that have been suggested

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

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

Johnson (1932) and Jeffreys (1948) have proposed statistically motivated approaches. 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.

The assumptions for this to hold are the most bland for $k = 1$, which was Jeffreys' suggestion. This is also an extremely common engineering approach, as it just calls for adding one to each r.

The IBM speech recognition group has found the formula introduced by Good (1953) and attributed to Turing to be a useful estimator in building a language model for speech recognition applications (Nàdas, 1984; Nàdas, 1985; Katz, 1987). Katz is responsible for the term ''backing-off,'' which we have adopted, to indicate the process of building an n-gram model based on an (n-1)-gram model. Katz suggested using the Good-Turing estimator to build from a unigram model to a bigram model and on to higher n-grams. We have adopted this backing-off approach of extending an n-gram model to an (n+1)-gram model, although the methods we study are novel. Katz also used the following two approximations of the Good-Turing method, motivated by constraints on processing capability. First, for $r > k$ (specifically, $k = 5$), set $r^* = r$. Second, treat the bigrams observed just once as if they were unobserved.

2. Selection of materials for testing estimation methods

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 $N = 4.4 \times 10^7$ words in the corpus. The AP wire may not be sufficiently representative of English as a whole to use bigram frequencies derived from it in applications. However, as a means of testing methodologies, a large corpus size is desirable, and it is much larger than alternatives such as the Brown Corpus.

We split the 1988 AP wire into two portions, one for estimating probabilities and one for testing the estimates, and simply consider our universe to be the 1988 AP wire, rather than English. 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 in bigram frequencies 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 IBM speech recognizer (Nàdas, 1984).

Our goal is to develop a methodology for extending an n-gram model to an (n+1)-gram model. We regard the model for unigrams as completely fixed before beginning to study bigrams. This includes specifying the vocabulary, $\mathbf{V}$, 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.

3. The Basic Good-Turing and Cat-Cal Methods

This paper studies two alternatives to the MLE, which we call Good-Turing (GT) estimates and Cat-Cal (CC) estimates. This section defines what we mean by ''basic'' variants of each of these methods.

The Good-Turing estimator may be familiar to some readers from its applications in population biology. The key insight suggested by Turing and developed by Good (1953), is the use of N_r, the number of bigrams which occur r times. We may refer to N_r as the frequency of frequency r. The entire distribution $\{N_0, N_1, N_2, \cdots\}$ is available in addition to the particular r. The Turing estimator uses this extra information as $r^* = (r+1) N_{r+1} / N_r$. Good suggested smoothing the observed N_r to provide a sequence $\{\mathbf{S}(N_r)\}$ for use in the formula, and discussed a number of possible smoothing approaches. We call r^* a basic Good-Turing estimator (BGT) when smoothed N_r's are used. Many different basic Good-Turing estimators are possible depending on how the smoothing is performed. A derivation of Turing's formula is given in Appendix A. The proof there shows that the Good-Turing equation rests on the assumption that the distribution of each bigram is binomial.

The following example may clarify the use of the Good-Turing estimator. The first column of the table shows frequencies of bigrams from zero through nine. (For this example, we are using small frequencies, r, with large N_r so that we do not need to discuss smoothing the N_r's.) This column is also the MLE, of course. The second column shows N_r, how many bigrams had those frequencies in a training sample of about twenty two million bigrams, which was half of the 1988 AP wire. For instance, 2,053,146 bigrams appeared exactly once in the sample. The third column shows the adjusted frequencies as calculated by the Turing formula.

Table 1
Basic Good-Turing Method
Combined Frequencies of Bigrams

r (=MLE)	N_r	r^*(=BGT)
0	160,519,590,316	0.0000128
1	2,053,146	0.446
2	458,136	1.26
3	191,809	2.24
4	107,522	3.25
5	69,883	4.19
6	48,809	5.21
7	36,345	6.21
8	28,201	7.28
9	22,821	

The adjusted frequencies, r^*, can be compared to the raw frequencies, r. They have the same order, and do not differ greatly. The method assigns some probability to bigrams which have not been seen, suggesting that we should act as if we had seen each of them 0.0000128 times instead of zero times. In order to compensate for moving 160,519,590,316 bigrams from 0 to 0.0000128, some larger frequencies must be adjusted downwards. In this case, the adjusted frequencies, r^*, for all observed bigrams are less than the corresponding raw frequencies, r.

Notice that the calculations for $r = 0$ rests on knowing N_0, the number of bigrams that we have not seen. We can calculate this because we know the vocabulary size, V, from the unigram model. (This marks a great difference in our application of the Good-Turing formula from the kinds of applications made in population biology, where inferences about the population size are the desideratum.) The total universe of bigrams that we wish to know about has size $V^2 \approx 1.6\times 10^{11}$. N_0 is the difference between V^2 and the number of distinct bigrams seen, $\sum_{r>0} N_r$. Note that $N_0 \approx V^2$ since $N_0 = V^2 - \sum_{r>0} N_r$ and $\sum_{r>0} N_r < N << V^2$. In other words, hardly any of the possible bigrams have been seen.

The GT estimate improves on MLE by making use of more information, namely $\{N_r\}$. The Cat-Cal estimate gathers even more information. The training text is divided into two halves. *Categorize* each bigram, b, by its observed frequency $r_1(b)$ in the first part of the text. Denote the number of distinct bigrams in the category by $N_r = \sum_{b | r_1(b) = r} 1$. *Calibrate* the category by counting all occurrences of all the bigrams in the category in the second part of the text, $C_r \equiv \sum_{b | r_1(b) = r} r_2(b)$, where the $r_2(b)$ is the observed frequency of the bigram, b, in the second half. The adjusted frequency to use for the category is $r^* = C_r / N_r$. Again, when applied to all the data, we refer to this as the ''basic'' Cat-Cal method (BCC). The only assumption behind this method is that the same process generates both halves of the text. This assumption is much weaker than the binomial assumption of GT.

The following example may clarify the calculations in the Cat-Cal method. The first two columns are as in the previous table. The third column gives the extra information collected for the Cat-Cal method, C_r, and the fourth column gives the basic Cat-Cal estimates, C_r / N_r.

Table 2
Basic Cat-Cal Method
Combined Frequencies of Bigrams

r	N_r	C_r	BCC	repeat	Σr_2^2	variance
0	160,519,590,316	2,046,125	.0000127	.0000127	2,980,905	.0000185
1	2,053,146	919,645	.448	.448	2,069,343	.808
2	458,136	577,518	1.26	1.26	1,865,654	2.48
3	191,809	431,839	2.25	2.25	1,831,325	4.48
4	107,522	347,424	3.23	3.22	1,805,150	6.36
5	69,883	293,953	4.22	4.23	1,827,811	8.41
6	48,809	257,141	5.20	5.22	1,858,543	10.5
7	36,345	223,574	6.20	6.19	1,832,738	12.4
8	28,201	205,171	7.22	7.25	1,898,443	14.5

The adjusted frequencies for the BCC can be compared to the adjusted frequencies for the BGT as well as to the MLE. The differences between the BCC and the BGT are limited to the third significant figure, while the differences of either from the MLE are in the first significant figure.

We established the Cat-Cal method originally as a standard against which to compare other methods. However, we came to realize that it could itself be used as a practical method. Thus Cat-Cal plays two roles, standard and potential method. We can evaluate Cat-Cal as a potential method by comparing it to a different replication of Cat-Cal as a standard. The fifth column of the table contains the results of repeating the basic Cat-Cal method after exchanging the texts used for categorization and calibration. The differences are again limited to the third significant figure, showing that BCC as a potential method agrees well with our standard.

The sixth column shows the extra information required to calculate the variance of the observed $r_2(b)$ for $b \mid r_1(b) = r$. Calculating this variance, shown in the last column, makes clear that the BCC is an *estimate* of some underlying population parameter. The greatest interest is in the expected value of an estimate, but it is also important to characterize its uncertainty or variance. Note that the variances are approximately twice BCC. We will return to this point later.

4. The Enhanced Methods

A key suggestion of this work is the introduction of a second predictor of frequency of observation in addition to an observed frequency; accounting for the second predictor constitutes what we call an *enhanced* method. We study an enhanced Good-Turing method and an enhanced Cat-Cal method. Both enhanced methods allow us to *differentiate* among the many bigrams which have not been seen. We will show that about 1200 significantly different probabilities can be estimated for bigrams not seen in the training text. Thus the probabilities estimated by the enhanced Good-Turing method and the enhanced Cat-Cal method for the unseen bigrams are neither zero nor identical. We will refer to either method used without a second variable as a ‘‘basic’’ method.

The backing-off approach to a bigram model regards a bigram as a pair of unigrams (words). Supposing we had no empirical information on a bigram frequency, we might still hazard a guess as to the bigram’s probability based on characteristics of the words in the bigram. The basis for this guess becomes a second predictor when empirical frequency information *is* available. We build a bigram model by applying a basic method within each category of bigrams grouped by the second predictor. The application of Turing’s formula is valid in this case because the population of bigrams is prespecified; it depends only on the unigram model.

4.1 Unigram Estimates: a second predictor

A possible second predictor for bigrams is the following. If the text were composed of independently generated words, then the probability of a pair of words would just be the product of the probabilities of the individual words. Of course, the words are not independently generated, but the frequency predicted on this basis is still a useful predictor, as shown in the following figure.

A Second Useful Predictor of Bigram Frequencies

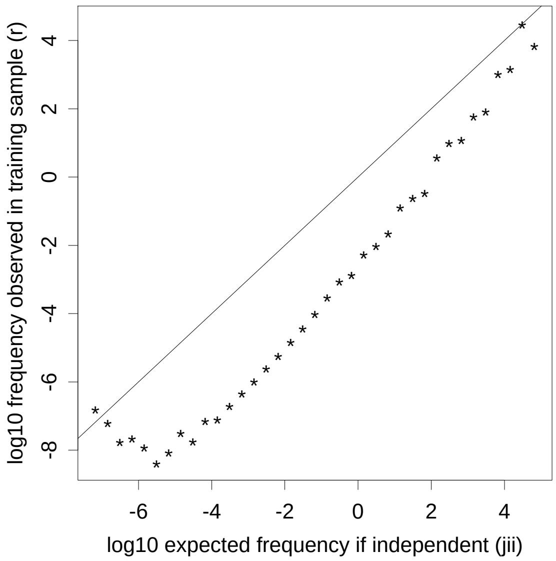

Figure 1. *Enhancement* of the Good-Turing or Cat-Cal methods consists of using the basic methods on categories defined by a second predictor. The second predictor used in this paper is the frequency with which we would expect to see each bigram if each word of the text were selected independently, which we denote here by jii, acronym for "joint if independent." Its logarithm (base 10) is shown as the abscissa. The ordinate is the average of observed frequencies for words grouped by jii. The figure shows that the average frequency is strongly correlated with the independent frequency over about ten orders of magnitude.

Let $jii = Ne(p(x))\,e(p(y))$ be the predicted frequency if the words, x and y, were independently distributed with the probabilities estimated by the unigram model as $e(p(x))$ and $e(p(y))$. Since the population probabilities, such as $p(x)$, are estimated by the unigram model, we refer to values of jii as "Unigram Estimates (UE)" when we compare them to other estimates, such as MLE or GT. We assume that the unigram model provides variances of the estimates as well as the estimates.

Other predictors could be used for the grouping; jii is one of many possibilities. We do not know what makes one variable better than another for grouping. A necessary property of the grouping variable is that it be possible to count the number of types included in each group, because we need to know N_0. We hypothesize that if one variable predicts r better than another, then it will make a better grouping variable. It is useful for smoothing that jii is a continuous variable.

4.2 Smoothing Methods

We have introduced two variables, r and jii, for predicting frequency observations. Practical methods may need smoothing across these variables. We find that both Cat-Cal and Good-Turing need smoothing across jii, while Good-Turing also needs smoothing across frequency, r.

The smoothing across jii is accomplished by controlling the size of the categories, or bins, derived from the continuous variable. There is a trade-off between specificity of jii and smoothness in selecting how many jii groups to use. Using fewer bins induces more smoothing; more bins less smoothing. We use about 35 groups, taking three groups in each factor of ten. We have not studied this trade-off more than to note that ten groups per order of magnitude of jii gave results that we kept feeling a need to smooth, so we went to three groups per order of magnitude.

After dividing all the bigrams into groups, we calculate N_{jii}, the number of distinct bigrams in each group, so that we can calculate the number of bigrams not seen. The calculation of the N_{jii} is made as follows. For each pair (r_1, r_2) of observed frequencies of the unigrams, determine $jii = r_1 r_2 / N$. >From that determine the bin for jii, and add the product of the number of distinct unigrams seen with the two frequencies to the cumulating count for that bin.

Let N_{jr} be the frequency of frequency r in jii bin j. The following figure is a plot of N_{jr} for r= 0, 1, 2, 3 as a function of jii.

Numbers of Distinct Bigrams

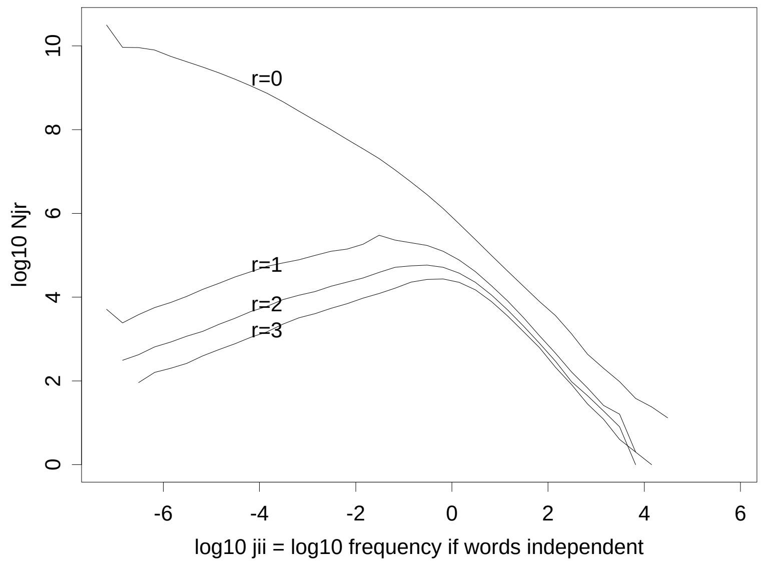

Figure 2. The bigrams are divided into groups according to the logarithm of jii, three groups per factor of ten. This figure plots N_{jr}, the number of distinct bigrams observed with frequency r within the jii bin j, for r= 0, 1, 2, 3 as a function of $\log_{10}(jii)$. Notice that N_{j0} is qualitatively different from N_{jr} for other r. We treat it separately in the following analysis.

For the smoothed values of N_{jr} required by the enhanced Good-Turing method, we treated $r = 0$ separately, and smoothed $\log(N_{j0})$ against $log(jii)$. For $r > 0$, we smoothed $log(N_{jr})$ against $\log(r)$.

The following figure shows logarithms of N_{jr} versus logarithms of r for $j = 22$. The non-zero raw values are plotted in the left panel. Since most of the N_{jr} are zero for large r, we need to account for them. Good used groups of frequencies. We average each non-zero N_{jr} with the zero N_{jr} that surround it: Order the non-zero N_{jr} by r, and let q, r, and t be successive indices of non-zero values. Then replace N_{jr} by $N_{jr} / .5(t - q)$. In other words, we estimate the expected N_{jr} by the *density* of N_{jr} for large r. For small r, there is no change because the length of the interval is one. For large r, the change can make a difference of up to four orders of magnitude. Figure 3 shows before and after plots for the 22nd group for which we would predict a frequency, jii, of 1.4 observations per year if the words were independent.

Averaging Zeros and Smoothing

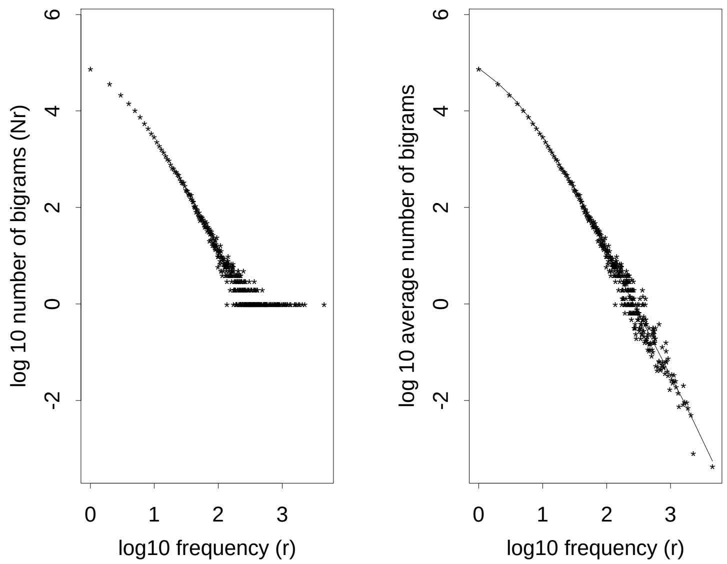

Figure 3. The left panel plots logarithms of raw non-zero N_{jr} against logarithms of $r > 0$, for a fixed bin ($jii \approx 1.4$). Each point represents one value of r. The right panel shows the raw N_{jr} averaged in with all the N_{jr} which are zero. The result is nearly linear over three orders of magnitude in r. We use a smoother that adjusts the amount of smoothing locally to take account of the change from low variance data for small r to high variance data for large r. The smoother also guarantees a smooth first derivative. The result of smoothing is shown by the solid line.

The averaging can be seen to restore the downward trend that is clear in the region with low quantization noise. The noise in the averaged data, and the sensitivity of the Good-Turing formula to slopes among the N_r or $\mathbf{S}(N_r)$ shows the necessity of a smooth for all but the smallest r. Without averaging and smoothing, most of the N_r are zero for large r. Thus, the calculation N_{r+1}/N_r would be grossly wrong.

The smoother we use, described by Shirey and Hastie (1988), has two useful properties for this application. First, it uses a technique called local cross validation to determine the range of data to smooth. The values for small r get very local smoothing, while for large r, nearly all the data is used. Second, the smoothness of the first derivative can be directly controlled, and we require it to be quite smooth. We make this requirement because we want to use $N_{j(r+1)}/N_{jr}$, which is essentially a derivative with respect to r.

5. Testing the Enhanced Good-Turing and Cat-Cal Methods

The assumptions of the Good-Turing theorem may seem innocuous: a finite set of types and a marginally binomial distribution. The first assumption (finiteness) was satisfied by construction. But the second assumption is not; we know that word sequences are not independent. There are considerable correlations within any discourse. Thus, we need to test how well a method assuming a marginally binomial distribution works in practice. The assumptions of the Cat-Cal method, that the two samples are from the same universe of discourse, also needs to be tested.

In this comparison, the Cat-Cal method plays two roles: standard of comparison, and potentially useful method. As a standard of comparison, we use a categorization corpus and a calibration corpus, each half of our total text. The categorization corpus is the same as the training corpus for the Good-Turing method. As a potentially useful method, comparisons with GT would be facilitated by splitting the training corpus into two parts. But our primary interest is is understanding the properties of each estimator, so we use the

Cat-Cal method interchanging the roles of calibration and categorization corpora, which doubles the sample sizes and decreases the noise for observing Cat-Cal.

In this section we will be comparing five estimates and their variances of bigram probabilities. These five estimators are the standard (S), the maximum likelihood estimator (MLE), the unigram estimator (UE), enhanced Cat-Cal (CC), and enhanced Good-Turing (GT). The section has three subsections in which we first compare the five estimates qualitatively, then introduce the variances of the estimates and compare these qualitatively, and finally make a delicate quantitative comparison.

5.1 Qualitative Evaluation of the Methods

We find that the Good-Turing estimates and the Cat-Cal estimates agree very well with the standard estimates over the entire range of data that we can test. The smallest frequency observations are the most critical. The following figure shows the results for $r = 1$.

Enhanced Good-Turing and Cat-Cal Agree with the Standard for r=1

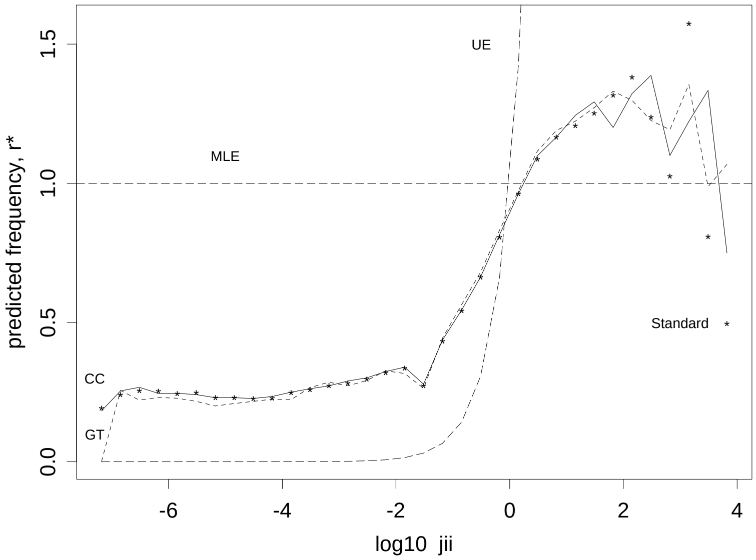

Figure 4. Five predicted frequencies are shown in this and following figures: (1) the standard, S, shown by points, (2) the maximum likelihood estimate, MLE, shown by long dashes, (3) the unigram estimate, UE, shown by long dashes, (4) the enhanced Cat-Cal estimate, CC, shown by a solid line, and (5) the enhanced Good-Turing estimate, GT, shown by short dashes. These estimates are plotted against the logarithm of the unigram estimator, jii. Note that CC and GT agree closely with the standard. They are quite distinct from either the MLE or UE but lie approximately between these two primary estimators.

The Cat-Cal and enhanced Good-Turing estimates can be viewed as combining the MLE and UE, and indeed, these ‘‘combined’’ estimates usually lie between the two long dashed lines.

Figure 5 presents a plot similar to Figure 4 for the important case of $r = 0$. Figure 6 below shows similar plots for $r = 2, 5, 8, 11, 14$, and 17.

Enhanced Good-Turing and Cat-Cal Agree with the Standard for r=0

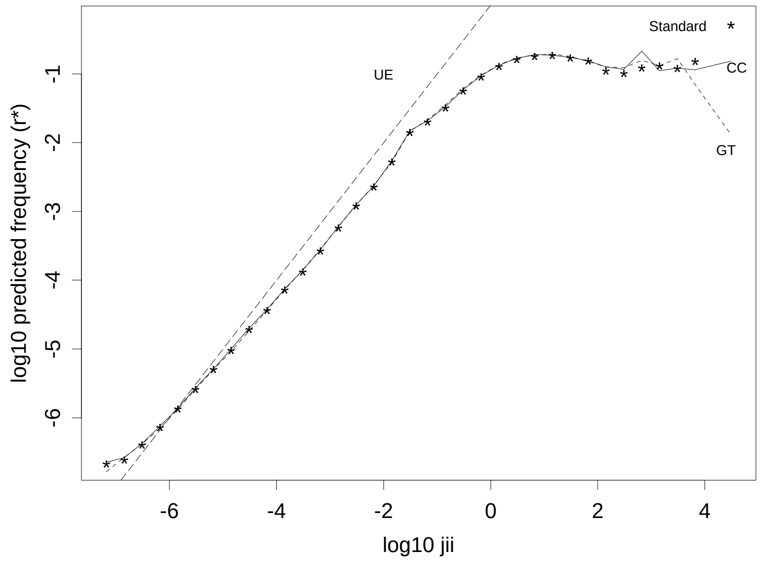

Figure 5. Note that CC and GT agree closely. They are quite distinct from the MLE of 0, which cannot be plotted on a logarithmic scale. They also depart substantially from UE.

For frequency zero, the range of CC and GT is about five orders of magnitude, four orders of magnitude larger than for any other frequency. Over this range, both GT and CC agree well with the standard estimates. At the resolution shown, there is no visible difference between the three estimates for most of the range.

Enhanced Good-Turing and Cat-Cal Agree with the Standard for Small r

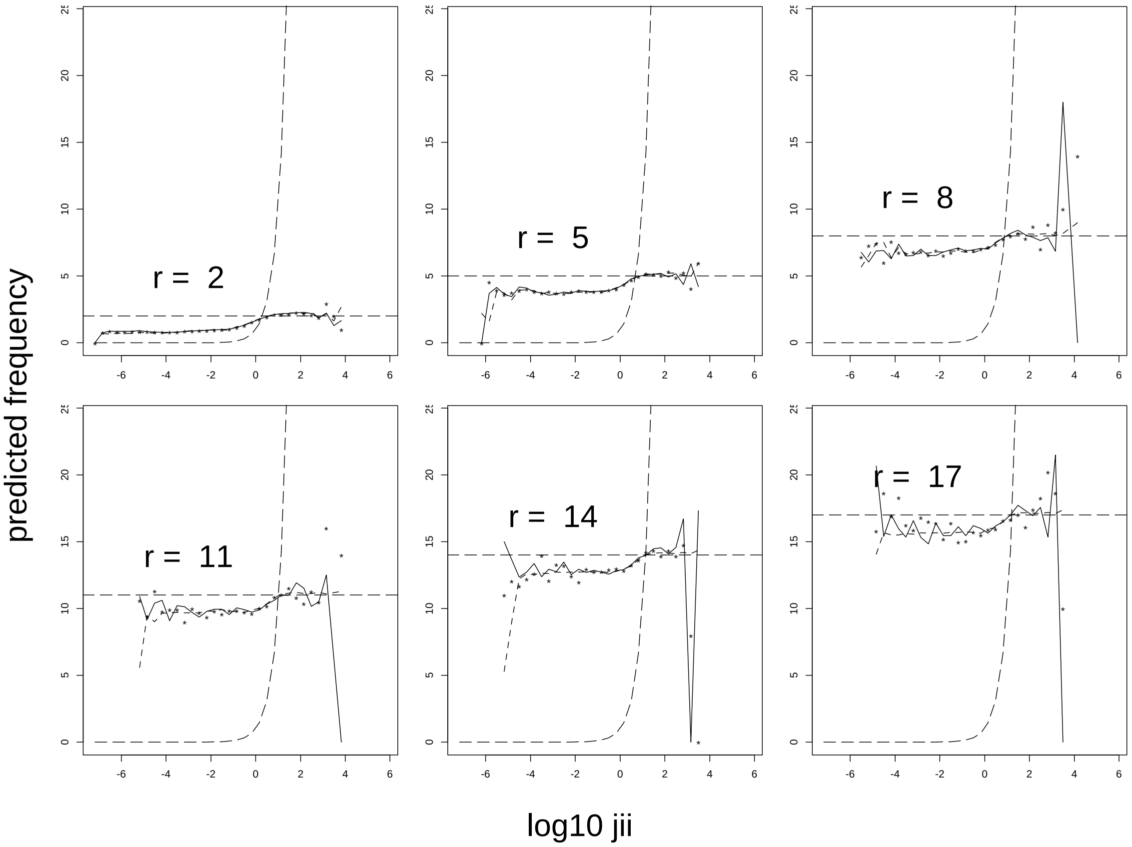

Figure 6. Both GT and CC are much better than MLE and UE. The difference is more important for small r.

Note that $r*$ depends more on jii when r is small; the slope of $r*$ is very steep for $r = 0$, and pretty flat for $r = 17$. This means that UE is more important when r is small. We will return to this when we consider the number of significantly different probabilities. After examining the variances of predicted frequencies in the next subsection, we will return to quantitative comparisons.

5.2 Qualitative Comparison of Variances Predicted

We said in Section 3 that the Cat-Cal method could be used to calculate variances. The calculation for v_{jr}^{CC}, the variance of observed frequencies about the mean for bigrams with frequency $= r$ and $jii = j$ is

$$v_{jr}^{CC} = \frac{\Sigma r_2^2(b)}{N_{jr}} - r*^2$$

where the sum is over bigrams, b, such that $r_1(b) = r$ and $jii = j$. This variance is observed by CC as a standard, and predicted by CC as a potential method. The Good-Turing theory also predicts the variance observed as the standard. Let v_{jr}^{GT} be the prediction. It is composed of (a) the variance of the true probabilities of of the bigrams in the group about their mean plus (b) the variance of observed frequencies for each group member about the their expectations. Appendix A derives the following expression for the sum of these two terms:

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

Both the MLE and the UE assume a binomial distribution for each bigram. Therefore the following calculation is made for each of them. The two parts of variance can be regarded as (a) measurement variance and (b) observation variance. For a single bigram, distributed binomially with probability p, the measurement variance and observation variance are each given by $Np(1-p) \approx Np = E(r)$. Thus the sum of the two is $2E(r)$, estimated as $2r^*$. The following figure compares the observed and predicted variances for $r = 1$.

Enhanced Good-Turing and Cat-Cal Variances Agree with the Standard for r=1

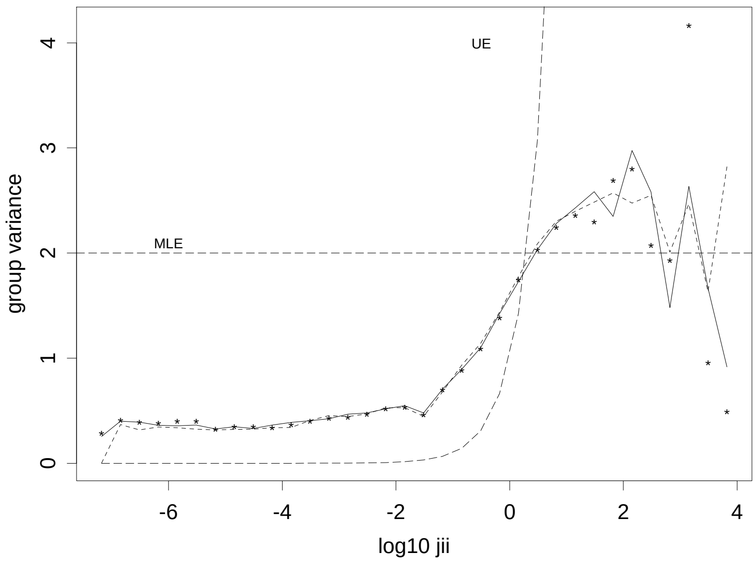

Figure 7. Observed or predicted variances for the group of bigrams with $r = 1$ are shown as a function of $log(jii)$. Again, the standard is shown with points, MLE and UE with long dashes, CC with a solid line, and GT with short dashed lines. GT and CC follow the standard closely showing reliable estimates of the variance in estimation and observation. Both are much closer to the standard than are MLE or UE.

Enhanced Good-Turing and Cat-Cal Variances Agree with the Standard for r=0

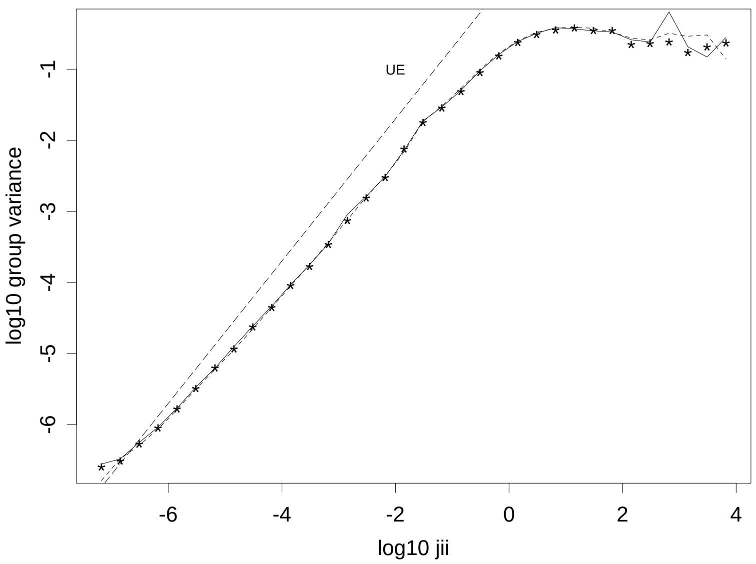

Figure 8. UE is too low for small jii and too large for large jii; MLE can't even be plotted. Predictions from GT and CC follow the observations closely.

Enhanced Good-Turing and Cat-Cal Variances Agree with the Standard for Small r

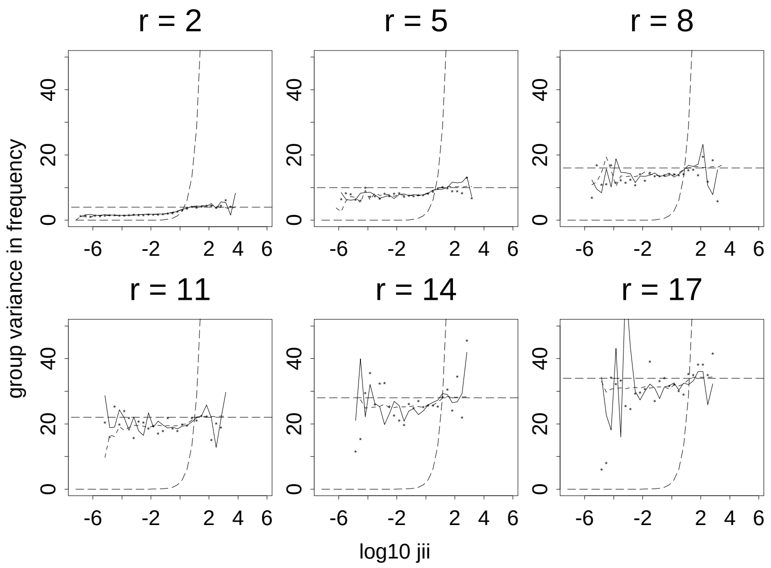

Figure 9. UE is too low for small jii and too large for large jii. MLE misses the variation with jii. GT and CC follow the standard closely.

The three figures in this section have shown that there is good qualitative agreement between the variances predicted by GT and CC and those observed as the standard. In the next section, we use GT variances to compare the differences of GT and CC from the standard estimates, since they are defined in more cells than are CC or S variances.

5.3 Quantitative Comparisons of Good-Turing and Cat-Cal Methods

Now that we have qualitatively useful variances, it is natural to compare other estimates to the standard estimates by comparing their difference to the variance: $t_{jr} = (r* - r^S_{jr})/\sqrt{v^{GT}_{jr}}$. 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^S_{jr}. As explained in the previous subsection, we use the Good-Turing variance, v^{GT}_{jr}, because it appears to match the Cat-Cal variance while being less noisy and defined in more cells.

We have some expectations about these t-scores. In the first place, a perfect predictor would give t-scores of about one, because the variance of one standard observation is used as the denominator. However, a replication of Cat-Cal is not such a perfect predictor, since *each* of the observations has the variance used in the denominator. Thus, for Cat-Cal as a practical method, we expect each of the terms to have the assumed variance, and the variance of their difference will then be about twice the assumed variance. Then the difference of the two terms will have variance twice that assumed and the t-score is expected to be about the square root of two. This assumes that the two samples are independent and generated by the same process. (Had we made CC estimates by splitting the training sample, the expected t-score would be the square root of five: one for the variance in the standard, and four for the variance in CC, as it would need to be multiplied by two to compensate for half the size of the training set.)

For small frequencies we will have about 35 t-scores, one for each jii bin. The following two figures show two sets of these t-scores, and a way to summarize them.

For r=3, Good-Turing Differs Significantly from the Standard
enhanced Cat-Cal is better than Good-Turing

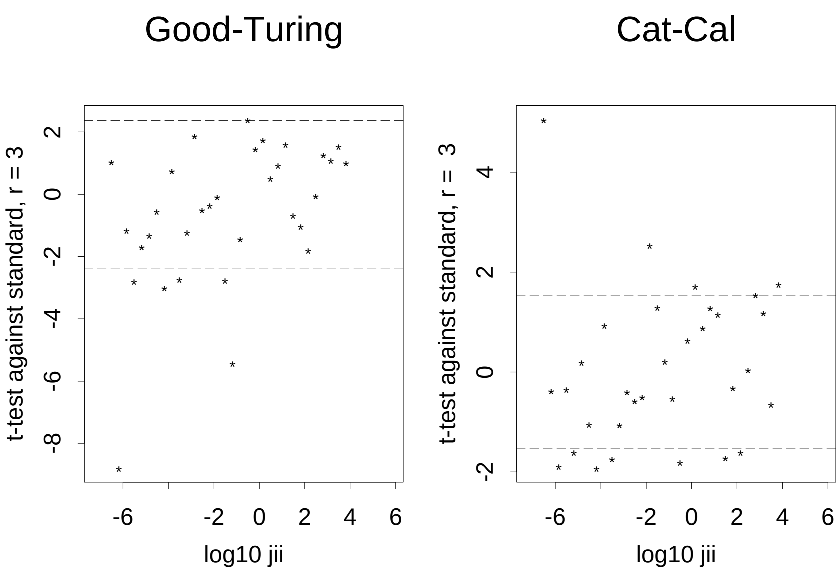

Figure 10. The left panel plots t-tests of the differences between GT and the standard estimates. The right panel plots t-tests of CC versus the standard. Dashed lines are drawn at plus and minus the RMS (root mean square) of the t-values plotted. It is clear here that the small differences between GT and the standard estimates are indeed significant, since an RMS t-value of about one would reflect chance differences. The differences between CC and the standard are not different from their chance level of 1.41.

The RMS (root mean square) t-score for GT is 2.4, well above the unity that a perfect predictor would achieve. The CC value, 1.52, is not significantly different from our expectation for it, 1.41. Another frequency, $r = 15$ is shown in the following figure.

**For r=15, the Differences from Standard are not Significant
enhanced Good-Turing is better than Cat-Cal**

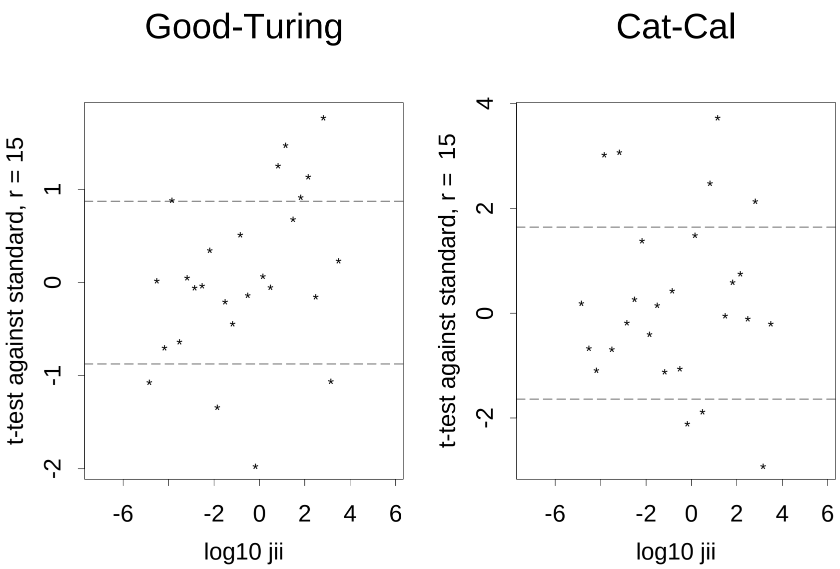

Figure 11. Both left and right panels plot t-scores, as in Figure 10, for $r = 15$. Neither panel shows significant differences from ideal performance. The variance in the second panel is higher, because two empirical observations have been made.

The RMS t-values are .95 (GT) and 1.55 (CC). These are not significantly different from our expectations for a perfect predictor and an ideal remeasurement, respectively. As a result, the enhanced Good-Turing estimates are now preferable to the Cat-Cal estimates. Since the t-scores vary by frequency, *r*, we present a plot showing their RMS values as a function of *r* in the following figure. Each point in Figure 12 summarizes a panel analogous to those in Figures 10 and 11. Dashed lines compare GT with the standard while solid lines compare CC with the standard.

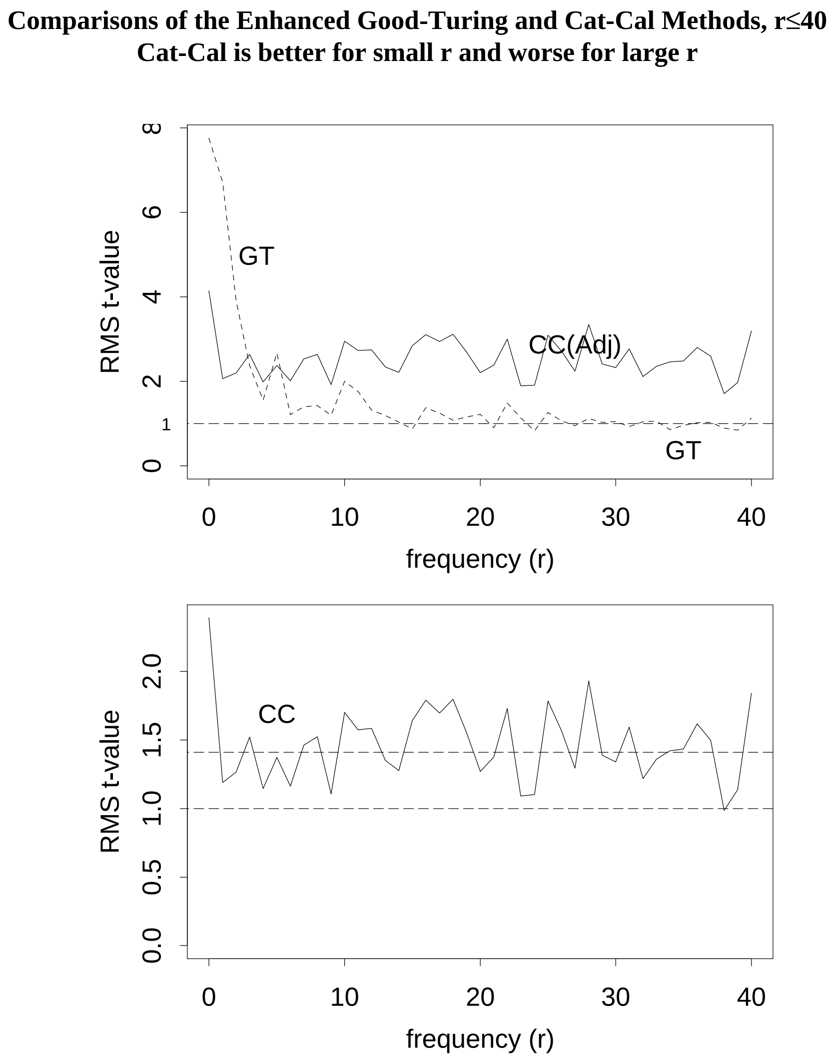

Figure 12. Each point on these plots represents one panel such as shown in the two previous figures. The plots show the RMS t-value for frequencies, r, from zero through 40. The solid lines compare CC with the standard; the short dashed lines compare GT with the standard. The best performance theoretically possible is an RMS error of 1, shown by a long dashed line in each panel. CC has a theoretical limit of $\sqrt{2}$, shown by the upper long dashed line in the lower panel, since it requires two empirical observations. The Cat-Cal estimates are not significantly different from their theoretical value, which validates the assumption that the two halves we selected have the same generation process. The CC values in the upper panel are adjusted for sample size to be comparable to GT values. GT begins worse than CC, becomes about equal for $4 \le r \le 10$, and is better for larger r.

We make two observations from this figure. First, for small frequencies[1] GT errors are larger than expected, while CC errors are as predicted. Second, for large frequencies, both GT and CC are behaving as predicted. GT is to be preferred when both behave as predicted, since the predicted behavior for GT is a t-score of one while that for CC on the same size sample is $\sqrt{3}$. However, CC is preferable for small frequencies. Our interpretation of this is that with fewer assumptions, Cat-Cal does better at frequency zero and achieves its ideal performance sooner than the Good-Turing method does. However, with stronger assumptions, GT will do better than CC when the assumptions are satisfied.

It is worth emphasizing that both of the methods proposed here, the enhanced Good-Turing and the enhanced Cat-Cal, are greatly superior to the MLE. The following figure makes this point by comparing the RMS t-scores for MLE and GT.

Comparing the Enhanced Good-Turing and MLE Methods
Good-Turing is better, especially when r is small

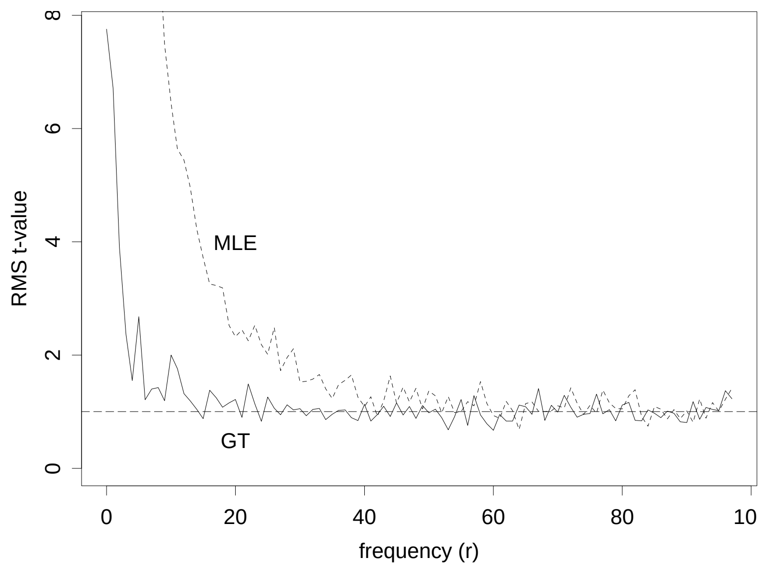

Figure 13. Each point on this plot represents one panel such as shown in Figures 10 and 11. The plot shows the RMS t-values for frequencies, r, from 0 through 100, provided they are less than the maximum RMS t-value for GT. The solid lines compare the enhanced Good-Turing estimates with the standard; dashed lines compare MLE with the standard. The best performance theoretically possible is a RMS error of 1, shown by the dashed line. The RMS t-score for the MLE (which begins about 250 for frequency zero) is distinctly greater than for the enhanced Good-Turing method through about frequency 40, and remains significantly greater over the range shown. In addition, recall that the MLE does not distinguish among bigrams with the same frequency, r. This is particularly problematic when $r = 0$. In the next section we show that the enhanced Good-Turing method is very good at distinguishing among bigrams that have not been seen ($r = 0$).

1. Zero frequency may be an exception to this generalization in that the CC RMS t-score at zero is 2.39, which is significantly different from 1.41. This still represents better performance than GT at zero frequency.

The MLE improves in performance over the range shown, but does not reach ideal performance within the range shown. Moreover, for frequencies less than about 40, the MLE is substantially worse than the Good-Turing method. The figure omits the MLE values for the smallest frequencies (for which the MLE has RMS t-values ranging from five to thirty times those of enhanced Good-Turing estimates) in order to show the comparison for larger frequencies more clearly.

6. How Many Significantly Different Probabilities?

In this section we show that estimates in adjacent jii bins differ quite significantly. This implies that interpolation is justified, and leads us to an estimate of the equivalent number of significantly different estimates.

For each jii, let $\hat{f}_{jr}$ denote a frequency estimated for jii bin $= j$ and frequency $= r$. The standard deviation of the difference, $(\hat{f}_{jr} - \hat{f}_{(j-1)r})$, is $\sqrt{\hat{v}_{jr}/N_{jr} + \hat{v}_{(j-1)r}/N_{(j-1)r}}$ where N_{jr} is the number of types for $jii = j$ and frequency $= r$, and $\hat{v}_{jr}$ are the Good-Turing variances for the frequencies predicted. Then the statistic we examine is the ratio

$$(\hat{f}_{jr} - \hat{f}_{(j-1)r}) \;/\; \sqrt{\hat{v}_{jr}/N_{jr} + \hat{v}_{(j-1)r}/N_{(j-1)r}}$$

This is a t-score for sequential differences in the estimates and is a test of the significance of the differences between adjacent jii bins. The following figure shows the t-scores for the particularly important case of $r = 0$, the bigrams not seen in the training sample.

About 1200 Significantly Different Probabilities for $r = 0$

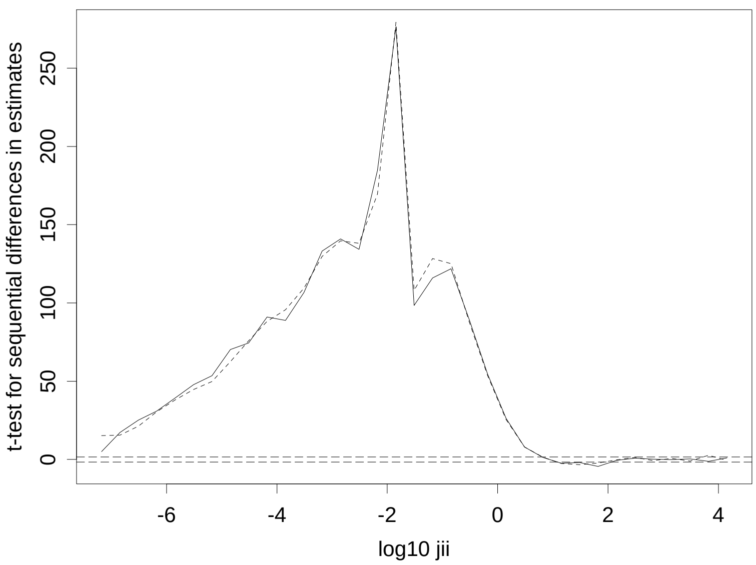

Figure 14. The horizontal axis shows the logarithm of jii. The vertical axis shows t-statistics for the difference between successive values of the Cat-Cal and Good-Turing estimates for bigrams not seen in the training text. The solid line show the t-statistics for CC; the short dashed line is for GT differences, and the long dashed lines are drawn at conventional significance levels of $\pm$ 1.65. These differences are highly significant, showing that interpolation between the observed values is justified. Doing so would give the equivalent of using about 1200 jii bins.

The figure shows that for many jii values, the differences are highly significant. We estimate the equivalent number of significantly different values by taking the sum of all the t-statistics and dividing by 1.65. For

$r = 0$, the equivalent number of significantly different values is 1245. The following figure shows the equivalent number of significant differences as a function of frequency.

Equivalent Number of Significantly Different Probabilities
we can distinguish bigrams with the same frequency very well for small frequencies

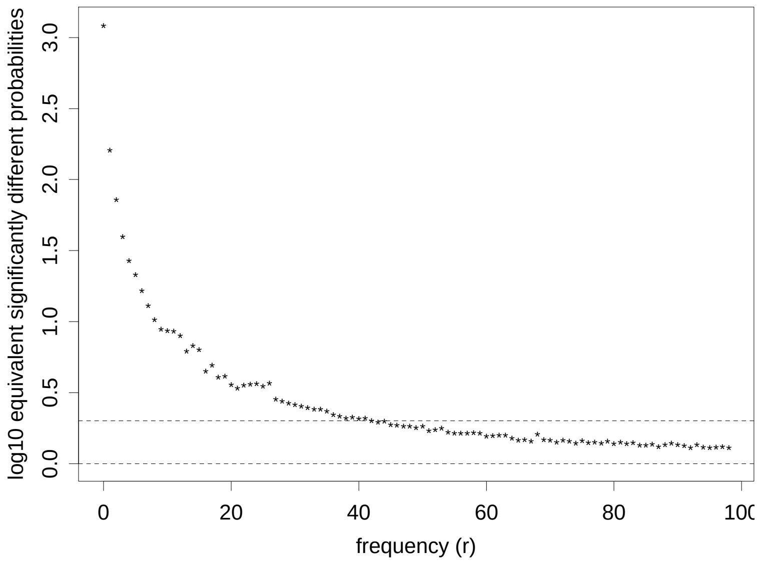

Figure 15. The plot shows $\log_{10}$ of the equivalent number of significantly different probability estimates for the group of bigrams observed r times. The dashed lines show the logarithms of one and two. While the number of significantly different values falls rapidly with increasing r, it remains above two through $r \approx 40$, and continues to be greater than one even through frequency 100. This range encompasses the majority of bigram tokens and indicates the value of a second predictor for practical applications.

The figure shows that the number of significantly different values falls rapidly from 1200 as r increases, remaining above 2, however, for $r \leq 40$. The decrease with increasing r is a consequence of the decreasing slope of of the lines shown in Figures 4, 5, 6, as r increases. The types of bigrams with frequency greater than 40 account for about 1.8 percent of types observed. Therefore, the splitting by jii is useful for about 98.2 percent of the *observed* bigram types and all of the unobserved bigram types. We conclude that enhancement is of considerable value for practical applications.

7. Sampling, Representativeness, and Extrapolation

We have shown that enhanced Good-Turing and enhanced Cat-Cal predictions agree well with an observed standard. However, the training and validation texts were chosen to minimize discrepancies from these methods' assumptions. This section argues that a training corpus needs to be carefully sampled to ensure representativeness, in order to make valid extrapolations. In support of this thesis we show that the language used in the first temporal half of our AP corpus is significantly different from the language used in the second half.

The following table shows basic Cat-Cal and basic Good-Turing predictions and an observed standard using the first half and last half of our AP corpus as training and validation texts.

Table 3
Basic Methods with First/Last Split Text:
the language use differs

r	S	SDS	BCC	BGT	tCC	tGT
0	0.00001684	.000000201	0.00001656	0.0001132	-143	-2730
1	0.4076	0.00105	0.4050	0.5259	-2.50	113.
2	1.0721	0.00352	1.0542	1.2378	-5.09	47.0
3	1.9742	0.00778	1.9714	2.2685	-0.36	37.8
4	2.8632	0.0123	2.8594	3.1868	-0.31	26.4
5	3.7982	0.0163	3.8087	4.2180	0.64	25.8
6	4.7822	0.0285	4.6860	5.2221	-3.37	15.4
7	5.7154	0.0294	5.7092	6.1839	-0.21	15.9
8	6.7330	0.0420	6.7261	7.1784	-0.17	10.6

The columns in this table are as follows: *r* is the frequency of bigrams in the training text. *S* is the standard Cat-Cal calculated with the first half as categorization and the last half as calibration. *SDS* is the standard deviation of *S*, calculated as V_r / N_r where V_r is the variance of the group of bigrams (column 7 in Table 2). *BCC* is the basic Cat-Cal calculated with the last half of the text for categorization and the first half for calibration. *BGT* is the basic Good-Turing estimate calculated from the first half. *tCC* is the t-score $(BCC - S)/SDS$. *tGT* is the t-score $(BGT - S)/SDS$.

A striking feature in the table is the systematic differences between BGT and S, reflected in highly significant t-scores for their differences. BCC also differs from S for $r \leq 2$. These observations are to be contrasted with the behavior when we use alternating pairs to split the corpus, as shown in the following table.

Table 4
Basic Methods with Alternating Bigram Split Text:
the language use is the same

r	S	SDS	BCC	BGT	tCC	tGT
0	0.0000127489	.0000001075	0.0000127421	0.000127858	-6.3	34.3
1	0.4476	0.000626	0.4483	0.44628	1.11	-2.13
2	1.2621	0.00233	1.2570	1.2560	-2.24	-2.64
3	2.2458	0.00481	2.2501	2.2423	0.89	-0.74
4	3.2278	0.00772	3.2232	3.2497	-0.60	2.84
5	4.2243	0.0110	4.2289	4.1906	0.41	-3.07
6	5.2040	0.0145	5.2173	5.2125	0.91	0.58
7	6.2005	0.0184	6.1934	6.2074	-0.39	0.37
8	7.2221	0.0224	7.2560	7.2830	1.51	2.71

Notice both BGT and BCC agree well with S when we split the corpus this way, although the fits are not as good for $r = 0$ as they are for larger *r*. Also notice that *SDS* is about half as large for this table. Since the N_r are of comparable magnitudes for the two splittings, this observation shows that the group variances are about four times as large for the first/last split as they are for the alternating bigram split.

We conclude from these observations that the language used in the first half of the AP corpus differs from the language used in the second half. If these two similar texts differ in language, then we can surely expect language differences at least as large between less related texts.

In other words, the methods studied here can be used to estimate bigram probabilities and their variances in a given universe of discourse, but they are accurate enough to show significant language differences between different texts. This raises two issues that must be faced for applications. First, the appropriate universe of discourse to sample from must be defined on the basis of the needs of the application. Second, in order to design a sample of text from an appropriate universe, it will be necessary to know how much variability there is between texts.

8. Conclusions

This paper has proposed two specific methods for backing-off bigram probability estimates to unigram probabilities: the enhanced Good-Turing method, and the Cat-Cal method. Three important points in this paper have extended the strength of these methods over previous methods:

- the use of a second predictor (e.g., *jii*) to exploit the structure of n-grams, the distinguishing feature between the *enhanced* Good-Turing method and the basic Good-Turing method.
- the estimation of variances for the bigram probabilities, which allows building significance tests for various practical applications, and in particular allows
- the use of refined testing methods that can show important qualitative differences even though quantitative differences may be small.

Section 3 introduced the Good-Turing methodology. Appendix A proves a theorem that formalizes Good's treatment. The theorem shows that the key assumption of the Good-Turing methods is that both training and test corpus are generated by a common marginally binomial distribution. While this is not a strong assumption, Cat-Cal, introduced in Section 5, is a method with even weaker assumptions. It only requires that the both training and test corpus are generated by the same process. Although we first came upon this method in searching for an appropriate way to test the Good-Turing estimate, we discovered that it too provides an estimate of r^* that may be as useful as the Good-Turing estimate.

It is important, however, to control very carefully for variations among different texts. It is simply not true that words or n-grams are independently distributed; probabilities vary systematically with time, topic, style, etc. We have found these factors can dominate the subtle differences that we have been trying to study, especially the difference between the Cat-Cal estimate and the Good-Turing estimate.

The Cat-Cal method was used in section 6 to verify the enhanced Good-Turing estimates. It is remarkable how close the two estimates agree over a wide range of r and *jii*. In contrast, a number of other popular estimates fail to do so well. For example, it is much worse to assume that $r^* = r$ (MLE) or that $r^* = jii$ (UE). The small differences between the Cat-Cal and enhanced Good-Turing methods favor Cat-Cal for $r \leq 6$ and enhanced Good-Turing for $r \geq 12$ in our tests.

The first point about these methods, the use of the second predictor, was discussed in section 4. The use of a second predictor is the basis on which we distinguish the enhanced Good-Turing method (GT) proposed here from the basic Good-Turing method and the enhanced Cat-Cal (CC) from a basic Cat-Cal. If we had not introduced a second predictor, all bigrams that were observed once would be considered equally likely, and all bigrams that were observed twice would also be considered equally likely, and so on. This is extremely undesirable. Note that there are a large number of bigrams that have been seen just once (2,053,146 in a training corpus of 22 million words); we do not want to model all of them as equally probable. Much worse, there are a very large number of bigrams that have not been seen (160,519 million bigrams in the same training corpus of 22 million words); we really do not want to model all of them as equally probable. By introducing the second predictor *jii* as we did, we were able to make much finer distinctions within groups of bigrams with the same number of observations r. In particular, for bigrams not seen in the training corpus, we have about 1200 significantly different estimates.

It would be interesting to consider other variables besides *jii*. One might consider, for example, the number of letters in the bigram. Katz (1987) proposes an alternative variable: the first word of the n-gram. Any variable that is not completely correlated with r would be of some use. *jii* has some advantages; it makes it possible to summarize the data so concisely that the relevant structure can be observed in a simple plot. Moreover, *jii* has a natural order and is continuous, so the number of bins can be adjusted for accuracy. In contrast, selecting the first word of the n-gram prescribes the number of bins.

In section 6.2, we discussed the second point we want to emphasize: estimation of variances. The methods introduced by Good allow one to calculate population variances under the same assumptions that one calculates probabilities. A few more lines of code allow the measurement of variances for each Cat-Cal category at the same time as one measures calibration probabilities. Variances are necessary to make statements about the statistical significance of differences between observed and predicted frequencies. In applications, such as (Church, Gale, Hanks, and Hindle, 1989), this will allow distinguishing unusual n-grams from chance variations. In our work, variances are used to test methods.

The third point we want to emphasize, the use of refined tests for differences in methods, is discussed in section 6.3. We compare four methods, MLE, UE, CC, and GT, to the standard. We calculate t-scores for the differences between the standard and a proposed method and aggregate results across jii. We find that the enhanced Good-Turing method has no bias when compared to Cat-Cal, and has a lower dispersion for frequencies above about 12. However, the marginally binomial assumption on which the Good-Turing methods rest is apparently not quite satisfied for small frequencies.

The language model presented here should be useful for preliminary work in disambiguation, but there are many ways that it could be improved. We have said very little about the unigram model; in fact, we have been using the MLE method to estimate the probabilities for the unigram model. One could apply the methodology developed here to improve greatly on this.

In closing, we ought to say a few words about sampling. Of course, a year of the AP Newswire is not a balanced sample of English. In Section 7, we mentioned some problems with extrapolating from one six month period of AP Newswire to another, let alone from one genre of English to another. For any particular application, one must be very careful to select an appropriate sample of text to use for training. The methods presented here will then allow accurate estimates of the bigram probabilities in replicates of the chosen text.

References

Cherry, L. L. (1981), ''Writing Tools - The STYLE & Diction Programs,'' Computing Science Technical Report 91, AT&T Bell Laboratories.

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, Gale, W., Hanks, P., Hindle, D., (1989) ''Parsing, Word Associations and Typical Predicate-Argument Relations,'' International Workshop on Parsing Technologies, CMU, August.

Church, K. W. and P. Hanks (1989) ''Word Association Norms, Mutual Information and Lexicography,'' ACL, Vancouver, Canada.

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

Kahan, S., Pavlidis, T., and Baird, H., (1987), ''On the Recognition of Printed Characters of any Font or Size,'' IEEE Transactions PAMI, pp. 274-287, March, 1987.

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

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.

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.

Nàdas, A., (1985), ''On Turing's formula for word probabilities,'' *IEEE Transactions on Acoustics, Speech, and Signal Processing,* v. ASSP-32 pp. 859-861.

Sproat, R. and Shih, C-L. (1990), ''A Statistical Method for Finding Word Boundaries in Chinese Text.'' *Computer Processing of Chinese and Oriental Languages*, to appear.

Appendix A
The Good-Turing Theorem

with J. B. Kruskal

Good (1953) gives an important formula and an argument for the formula, which has been widely used in biology. We state his result as a theorem and prove it in this appendix, in order to be able to see the assumptions of the formula.

Let s_ν, $\nu = 1, \cdots, S$ be a finite collection of *types* (for instance, words, bigrams, or species of animals). For each type, *tokens* (examples of words or bigrams, or animals) can be sampled. Let $B(N;\ p_1,\ \ldots,\ p_S)$ denote a sample of size N drawn from the S types, each type, s_ν, having a binomial distribution with probability p_ν. A multinomial distribution, for instance, would have such a marginally binomial distribution, but the theorem covers a broader class of distributions. Let N_r be the number of types whose frequency in a sample is *r*, and let r_ν be the frequency of the νth type.

Theorem: When two independent marginally binomial samples, $B_1(N;\ p_1,\ \ldots,\ p_S)$ and $B_2(N;\ p_1,\ \ldots,\ p_S)$ are drawn, the expected frequency, r^*, in the sample B_2 of types occurring *r* times in B_1 is

$$r^* = \frac{(r+1)}{1\ +\ 1/N}\,\frac{E(N_{r+1}\,|\,B(N+1;\ p_1,\ \ldots,\ p_S))}{E(N_r\,|\,B(N;\ p_1,\ \ldots,\ p_S))}$$

It immediately follows that

$$r^* \approx (r+1)\,\frac{E(N_{r+1}\,|\,B(N))}{E(N_r\,|\,B(N))}$$

For a practical computation, the expectations, $E(N_r\,|\,B(N))$ and $E(N_{r+1}\,|\,B(N))$, are estimated by smoothed values, $\mathbf{S}(N_r)$ and $\mathbf{S}(N_{r+1})$, giving

$$r^* \approx (r+1)\,\frac{\mathbf{S}(N_{r+1})}{\mathbf{S}(N_r)}$$

Proof of Theorem

The approximations both have relative errors of $1/N$. With $N \approx 10^7$, the approximations are very good. Thus, only the theorem needs to be proved.

We make three random choices independently and simultaneously. One is to choose one of the *S* types, using equal probabilities, $1/S$, for each type.[2] We include a subscript zero on expectations taken over this random choice. The two other choices generate random samples $B_1(N;\ p_1,\ \ldots,\ p_S)$ and $B_2(N;\ p_1,\ \ldots,\ p_S)$. We include subscripts 1 or 2 or both on expectations taken over these choices.

Let $\tilde{\mu}$ be the random variable which is the index of the type chosen in the first choice. Let $\tilde{R}_{\tilde{\mu}1}$ and $\tilde{R}_{\tilde{\mu}2}$ be the frequencies of the chosen type in B_1 and B_2 respectively.

We restate the theorem as

$$r^* \equiv E_{012}(\tilde{R}_{\tilde{\mu}2}\,|\,\tilde{R}_{\tilde{\mu}1}\ =\ r)$$

$$= \frac{(r+1)}{1\ +\ 1/N}\,\frac{E(N_{r+1}\,|\,B(N+1;\ p_1,\ \ldots,\ p_S))}{E(N_r\,|\,B(N;\ p_1,\ \ldots,\ p_S))}$$

2. Equal weights give a useful result. Other weights give other valid results, some of which may also be useful.

The proof relies on three lemmas:

1: $E_{012}(\tilde{R}_{\tilde{\mu}2} | \tilde{R}_{\tilde{\mu}1} = r) = NE_{01}(p_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r)$

2: $E_{01}(s_\nu = s_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) = \dfrac{p_\nu^r (1 - p_\nu)^{N-r}}{\sum_{\lambda=1}^{S} p_\lambda^r (1 - p_\lambda)^{N-r}}$

3: $E_N(N_r) = \binom{N}{r} \sum_{\lambda=1}^{S} p_\lambda^r (1 - p_\lambda)^{N-r}$

where E_N denotes expectation over any $B(N;\ p_1,\ \ldots,\ p_S)$ (either B_1, or B_2, in particular).

Then we have

$$
\begin{aligned}
r^* &= E_{012}(\tilde{R}_{\tilde{\mu}2} | \tilde{R}_{\tilde{\mu}1} = r) \\
&= NE_{01}(p_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) && \text{by lemma 1} \\
&= N \sum_{\nu=1}^{S} p_\nu E_{01}(s_\nu = s_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) \\
&= N \sum_{\nu=1}^{S} p_\nu \frac{p_\nu^r (1 - p_\nu)^{N-r}}{\sum_{\lambda=1}^{S} p_\lambda^r (1 - p_\lambda)^{N-r}} && \text{by lemma 2} \\
&= N \frac{\sum_{\nu=1}^{S} p_\nu^{r+1} (1 - p_\nu)^{N-r}}{\sum_{\lambda=1}^{S} p_\lambda^r (1 - p_\lambda)^{N-r}} \\
&= N \frac{E_{N+1}(N_{r+1}) / \binom{N+1}{r+1}}{E_N(N_r) / \binom{N}{r}} && \text{by lemma 3} \\
&= N \frac{\binom{N}{r} E_{N+1}(N_{r+1})}{\binom{N+1}{r+1} E_N(N_r)} \\
&= \frac{r+1}{1 + 1/N} \frac{E_{N+1}(N_{r+1})}{E_N(N_r)}
\end{aligned}
$$

which was to be proved.

Proof of Lemma 1

For $i = 1, \cdots, N$ and any $\nu \varepsilon \{1, \cdots, S\}$, let

$$\tilde{x}_{\nu i} = \begin{cases} 1 & \textit{if the ith token in sample 2 is of type } \nu \\ 0 & \textit{otherwise} \end{cases}$$

Then

$$\tilde{R}_{\tilde{\mu}2} = \sum_{i=1}^{N} \tilde{x}_{\tilde{\mu}i}$$

and

$$\begin{aligned} E_{012}(\tilde{R}_{\tilde{\mu}2} | \tilde{R}_{\tilde{\mu}1} = r) &= \sum_{i=1}^{N} E_{012}(\tilde{x}_{\tilde{\mu}i} | \tilde{R}_{\tilde{\mu}1} = r) \\ &= \sum_{i=1}^{N} E_{012}(p_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) \\ &= NE_{01}(p_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) \end{aligned}$$

which was to be proved.

Proof of Lemma 2

Let $f(\nu)$ be the event that $s_\nu = s_{\tilde{\mu}}$; $e(r)$ be the event that $s_{\tilde{\mu}}$ occurs r times; and $e(\nu,r)$ be the event that $s_\nu = s_{\tilde{\mu}}$ and s_ν occurs r times. Then $\{f(\nu) \& e(r)\} = e(\nu,r)$. Also

$$\bigcup_{\lambda=1}^{S} e(\lambda,r) = e(r)$$

and $\{e(\lambda,r)\}_{\lambda=1,\ldots,S}$ are mutually exclusive. Hence

$$\begin{aligned} E_{01}(s_\nu = s_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) = P(\nu = \tilde{\mu} | \tilde{R}_{\tilde{\mu}1} = r) &= P(f(\nu) | e(r)) \\ &= \frac{P(f(\nu) \& e(r))}{P(e(r))} = \frac{P(e(\nu,r))}{P(e(r))} \\ &= \frac{P(e(\nu,r))}{\sum_{\lambda} P(e(\lambda,r))} \end{aligned}$$

Now, since $\tilde{\mu}$ is chosen with probability $1/S$ and r_ν is binomially distributed

$$P(e(\nu,r)) = \frac{1}{S} \binom{N}{r} p_\nu^r (1 - p_\nu)^{N-r}$$

so that

$$E_{01}(s_\nu = s_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r) = \frac{\frac{1}{S}\binom{N}{r} p_\nu^r (1 - p_\nu)^{N-r}}{\sum_\lambda \frac{1}{S}\binom{N}{r} p_\lambda^r (1 - p_\lambda)^{N-r}}$$

$$= \frac{p_\nu^r (1 - p_\nu)^{N-r}}{\sum_\lambda p_\lambda^r (1 - p_\lambda)^{N-r}}$$

which was to be shown.

Proof of Lemma 3

For any r,

$$N_r = \sum_{\nu=1}^{S} \delta(r, r_\nu)$$

where $\delta(r, f)$ is the delta function

$$\delta(r,f) = \begin{cases} 1 & r = f \\ 0 & r \neq f \end{cases}$$

Therefore

$$\begin{aligned} E(N_r | B(N;\, p_1,\, \dots,\, p_S)) &= \sum_\nu \delta(r, r_\nu) P(r_\nu | B(N)) \\ &= \sum_\nu P(r | B(N)) \\ &= \sum_\nu \binom{N}{r} p_\nu^r (1 - p_\nu)^{N-r} \\ &= \binom{N}{r} \sum_\nu p_\nu^r (1 - p_\nu)^{N-r} \end{aligned}$$

as was to be proved.

Predicted Variance in Observed Frequencies

The variance in observed frequencies results from (1) variance in population probabilities giving a variance to the expected frequencies, and (2) variance of observed frequencies about expected frequencies.

The variance of expected frequencies is just N^2 times the variance of population probabilities. The variance of the population probabilities is given by

$$E_{01}(p_{\tilde{\mu}}^2 | \tilde{R}_{\tilde{\mu}1} = r) - E_{01}^2(p_{\tilde{\mu}} | \tilde{R}_{\tilde{\mu}1} = r)$$

The second of these terms is r^*/N as proved in the theorem. Analogous techniques can be used to prove that the first term is $(r+1)^* r^*/N^2$, as was shown in Good (1953). Thus the variance in expected frequencies is

$$\begin{aligned} N^2 (\ (r+1)^* r^*/N^2 - (r^*/N)^2\ ) \\ = r^* (\ (r+1)^* - r^*\ ). \end{aligned}$$

The distribution of differences between observed and expected frequencies is a superposition of S binomial distributions. The vu^{th} distribution has variance $Np_\nu(1-p_\nu) \approx Np_\nu$. The composite therefore has variance approximately

$$\frac{1}{S}\sum_{\nu=1}^{S} Np_\nu = E_{01}(Np_\nu \mid \tilde{R}_\nu = r) = r^*.$$

The total variance predicted for the observed frequencies is thus

$$\begin{aligned} & r^*(\ (r+1)^* - r^*\ ) + r^* \\ & \quad = r^*(\ 1 + (r+1)^* - r^*\ ). \end{aligned}$$

Appendix B
Nomenclature

We have referred to the following estimators:

- Standard, a Cat-Cal estimate using the training corpus for categorization and a second validation corpus for calibration. Shown in figures by points.
- MLE, maximum likelihood estimate, $r* = r$, shown in figures by long dashed lines.
- UE, unigram estimate, $r* = jii$, shown in figures by long dashed lines
- BCC, basic Cat-Cal estimate, $r* = C_r / N_r$
- CC, enhanced Cat-Cal estimate, $r* = C_{jr} / N_{jr}$, shown in figures by solid lines
- BGT, basic Good-Turing estimate, $r* = (r+1) \frac{\mathbf{S}(N_{r+1})}{\mathbf{S}(N_r)}$
- GT, enhanced Good-Turing estimate, $r* = \frac{\mathbf{S} N_{jr+1}}{\mathbf{S} N_{jr}}$, shown in figures by short dashed lines

Other common notations are:

N, number of tokens in the training corpus

r, frequency of a bigram in training corpus

$r*$, estimated frequency of a bigram in a second corpus of size N

V, vocabulary size, number of types of unigrams in the training corpus

$S = V^2$, number of types of bigrams possible

N_r, number of bigrams with frequency r in the training corpus. Note that $\sum_r N_r = V^2 \;\; => \;\; N_0 = V^2 - \sum_{r>0} N_r$

$p(x)$, hypothesized population probability of a word, x

$e(p(x))$, unigram model's estimate of $p(x)$

$jii = Ne(p(x))\,e(p(y))$, frequency of bigram *xy* if the words occurred independently with the probabilities estimated by the unigram model

N_{jii}, the number of bigrams in a given jii bin, that is, bigrams within a range of 1/3 order of magnitude in jii

N_{jr}, the number of bigram with frequency $= r$ and jii bin $= j$

C_r, the observed count in a second corpus of bigrams with frequency $= r$ in the training corpus

C_{jr}, the observed count in a second corpus of bigrams with frequency $= r$ and jii bin $= j$
%!PS-Adobe-2.0 %%Copyright: Copyright (c) 1993 AT&T, All Rights Reserved %%Version: 3.4 %%DocumentFonts: (atend) %%Pages: (atend) %%EndComments /DpostDict 200 dict def DpostDict begin % % Copyright (c) 1993 AT&T, All Rights Reserved % % Version 3.4 prologue for troff files. %

/#copies 1 store /Prologue (dpost.ps) def /aspectratio 1 def /formsperpage 1 def /landscape false def /linewidth .3 def /magnification 1 def /margin 0 def /orientation 0 def /resolution 720 def /rotation 1 def /xoffset 0 def /yoffset 0 def

/roundpage true def /useclippath true def /pagebbox [0 0 612 792] def

/R /Times-Roman def /I /Times-Italic def /B /Times-Bold def /BI /Times-BoldItalic def /H /Helvetica def /HI /Helvetica-Oblique def /HB /Helvetica-Bold def /HX /Helvetica-BoldOblique def /CW /Courier def

/CO /Courier def /CI /Courier-Oblique def /CB /Courier-Bold def /CX /Courier-BoldOblique def /PA /Palatino-Roman def /PI /Palatino-Italic def /PB /Palatino-Bold def /PX /Palatino-BoldItalic def /Hr /Helvetica-Narrow def /Hi /Helvetica-Narrow-Oblique def /Hb /Helvetica-Narrow-Bold def /Hx /Helvetica-Narrow-BoldOblique def /KR /Bookman-Light def /KI /Bookman-LightItalic def /KB /Bookman-Demi def /KX /Bookman-DemiItalic def /AR /AvantGarde-Book def /AI /AvantGarde-BookOblique def /AB /AvantGarde-Demi def /AX /AvantGarde-DemiOblique def /NR /NewCenturySchlbk-Roman def /NI /NewCenturySchlbk-Italic def /NB /NewCenturySchlbk-Bold def /NX /NewCenturySchlbk-BoldItalic def /ZD /ZapfDingbats def /ZI /ZapfChancery-MediumItalic def /S /S def /S1 /S1 def /GR /Symbol def

/inch {72 mul} bind def /min {2 copy gt {exch} if pop} bind def

/setup { counttomark 2 idiv {def} repeat pop

landscape {/orientation 90 orientation add def} if /scaling 72 resolution div def linewidth setlinewidth 1 setlinecap

pagedimensions xcenter ycenter translate orientation rotation mul rotate width 2 div neg height 2 div translate xoffset inch yoffset inch neg translate margin 2 div dup neg translate magnification dup aspectratio mul scale scaling scaling scale

addmetrics 0 0 moveto } def

/pagedimensions { useclippath userdict /gotpagebbox known not and { /pagebbox [clippath pathbbox newpath] def roundpage currentdict /roundpagebbox known and {roundpagebbox} if } if pagebbox aload pop 4 -1 roll exch 4 1 roll 4 copy landscape {4 2 roll} if sub /width exch def sub /height exch def add 2 div /xcenter exch def add 2 div /ycenter exch def userdict /gotpagebbox true put } def

/landscapepage { landscape not { 0 height scaling div neg translate % not quite 90 rotate } if } bind def

/portraitpage { landscape { width scaling div 0 translate % not quite -90 rotate } if } bind def

/addmetrics { /Symbol /S null Sdefs cf /Times-Roman /S1 StandardEncoding dup length array copy S1defs cf } def

/pagesetup { /page exch def currentdict /pagedict known currentdict page known and { page load pagedict exch get cvx exec } if } def

/decodingdefs [{counttomark 2 idiv {y moveto show} repeat} {neg /y exch def counttomark 2 idiv {y moveto show} repeat} {neg moveto {2 index stringwidth pop sub exch div 0 32 4 -1 roll widthshow} repeat} {neg moveto {spacewidth sub 0.0 32 4 -1 roll widthshow} repeat} {counttomark 2 idiv {y moveto show} repeat} {neg setfunnytext}] def

/setdecoding {/t decodingdefs 3 -1 roll get bind def} bind def

/w {neg moveto show} bind def /m {neg dup /y exch def moveto} bind def /done {/lastpage where {pop lastpage} if} def

/f { dup /font exch def findfont exch dup /ptsize exch def scaling div dup /size exch def scalefont setfont linewidth ptsize mul scaling 10 mul div setlinewidth /spacewidth ( )

stringwidth pop def } bind def

/changefont { /fontheight exch def /fontslant exch def currentfont [1 0 fontheight ptsize div fontslant sin mul fontslant cos div fontheight ptsize div 0 0] makefont setfont } bind def

/sf {f} bind def

/cf { dup length 2 idiv /entries exch def /chtab exch def /newencoding exch def /newfont exch def

findfont dup length 1 add dict /newdict exch def {1 index /FID ne {newdict 3 1 roll put}{pop pop} ifelse} forall

newencoding type /arraytype eq {newdict /Encoding newencoding put} if

newdict /Metrics entries dict put newdict /Metrics get begin chtab aload pop 1 1 entries {pop def} for newfont newdict definefont pop end } bind def

% % A few arrays used to adjust reference points and character widths in some % of the printer resident fonts. If square roots are too high try changing % the lines describing /radical and /radicalex to, % % /radical [0 -75 550 0] % /radicalex [-50 -75 500 0] % % Move braceleftbt a bit - default PostScript character is off a bit. %

/Sdefs [/bracketlefttp [201 500] /bracketleftbt [201 500] /bracketrighttp [-81 380] /bracketrightbt [-83 380] /braceleftbt [203 490] /bracketrightex [220 -125 500 0] /radical [0 0 550 0] /radicalex [-50 0 500 0] /parenleftex [-20 -170 0 0] /integral [100 -50 500 0] /infinity [10 -75 730 0]] def

/S1defs [/underscore [0 80 500 0] /endash [7 90 650 0]] def