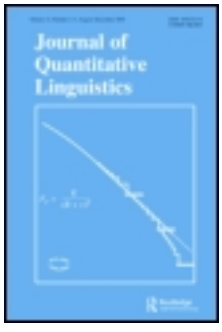

Journal of Quantitative Linguistics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/njql20

Good-turing frequency estimation without tears

William A. Gale [a] & Geoffrey Sampson [b]
[a] AT&T Bell Laboratories, USA
[b] School of Cognitive and Computing Sciences, University of Sussex, Falmer, Brighton, BN1 9QH, England Phone: +44 1273 678525 Fax: +44 1273 678525 E-mail:
Version of record first published: 21 Jul 2008.

To cite this article: William A. Gale & Geoffrey Sampson (1995): Good-turing frequency estimation without tears , Journal of Quantitative Linguistics, 2:3, 217-237

To link to this article: http://dx.doi.org/10.1080/09296179508590051

Journal of Quantitative Linguistics
1995, Vol. 2, No. 3, pp. 217-237

0929-6174/95/0203-0217$6.00

Good-Turing Frequency Estimation Without Tears*

William A. Gale and Geoffrey Sampson
AT&T Bell Laboratories, USA and University of Sussex, U.K.

ABSTRACT

Linguists and speech researchers who use statistical methods often need to estimate the frequency of some type of item in a population containing items of various types. A common approach is to divide the number of cases observed in a sample by the size of the sample; sometimes small positive quantities are added to divisor and dividend in order to avoid zero estimates for types missing from the sample. These approaches are obvious and simple, but they lack principled justification, and yield estimates that can be wildly inaccurate. I.J. Good and Alan Turing developed a family of theoretically well-founded techniques appropriate to this domain. Some versions of the Good-Turing approach are very demanding computationally, but we define a version, the Simple Good-Turing estimator, which is straightforward to use. Tested on a variety of natural-language-related data sets, the Simple Good-Turing estimator performs well, absolutely and relative both to the approaches just discussed and to other, more sophisticated techniques.

1. THE USE OF GOOD-TURING TECHNIQUES

Consider a population made up of individuals drawn from a large number of distinct species having diverse frequencies, including a few very common species, many rare species, and intermediate numbers of species of intermediate frequencies. We want to estimate the frequencies of the species in the population by counting their incidence in a finite sample. The "species" might be fauna or flora; but, in a linguistic context, they could be words (that is, word-types, represented in a sample of language by different numbers of word-tokens), word classes, bigrams (pairs of adjacent words), syllables, grammatical constructions, or the like. (In a linguistic context the terms "type" and "token" might seem more appropriate than "species" and "individual", but "type" is a relatively ambiguous word and we shall use "species" for the sake of explicitness.)

Say that the sample contains N individuals, and that for each species i it includes r_i examples of that species. (The number s of distinct species in the population may be finite or infinite, though N – and consequently the number of distinct species represented in the sample – must be finite.) We call r_i the *sample frequency* of i, and we want to use it in order to estimate the *population frequency* p_i of i, that is the probability that an individual drawn at random from the population will be a case of species i. Note that sample frequencies are integers from the range 0 to N, whereas population frequencies are probabilities, i.e. real numbers from the range 0 to 1.[1]

A very obvious method of estimating the population frequency is to divide sample frequency by size of sample, that is to estimate p_i as r_i/N. This is known as the *maximum likelihood estimator* for population frequency.[2] The maximum likelihood estimator has a large drawback: it estimates the population frequency of any spe-

* Address correspondence to Geoffrey Sampson, School of Cognitive and Computing Sciences, University of Sussex, Falmer, Brighton BN1 9QH, England, e-mail: geoffs@cogs.susx.ac.uk, tel.: +44 1273 678525, fax: +44 1273 671320.
The authors are very grateful to Professor I.J. Good for detailed comments on a draft of this paper. Responsibility for the contents of the paper is the authors' alone.
W.A. Gale has retired, March 1995.

cies that happens to be missing from the sample – any *unseen species* – as zero. If the population contains many rare species, it is likely that quite a number of them will be absent from a particular sample; since even a rare species has some positive population frequency, the maximum likelihood estimator clearly underestimates the frequencies of unseen species, and correspondingly it tends to overestimate the frequencies of species which are represented in the sample. Thus the maximum likelihood estimator is quantitatively inaccurate. More importantly, any estimator which gives zero estimates for some positive probabilities has specially unfortunate consequences for statistical calculations. These often involve multiplying estimated probabilities for many simple phenomena to reach overall figures for the probability of interesting complex phenomena; zeros propagate through such calculations, so that phenomena of interest are often assigned zero probability even when most of their elementary components are very common and the true probability of the complex phenomenon is reasonably high.

The second problem is often addressed by adding some small figure k to the sample frequencies for each species before calculating population frequencies: thus the estimated population frequency of species i would be $(r_i + k)/(N+sk)$. This eliminates zero estimates: an unseen species is assigned the estimated frequency $k/(N+sk)$. We shall call this the *additive* method. The additive method was advocated as an appropriate technique by Lidstone (1920: 185), Johnson (1932: 418-419), and Jeffreys (1948: §3.23), the first and third of these using the value $k = 1$. When the additive method is applied with the value $k = 1$, as is common, we shall call it the *Add-One* estimator. In language research Add-One was used for instance by the Lancaster corpus linguistics group (Marshall, 1987: 54), and by Church (1989).

But, although the additive approach solves the special problem about zeros, it is nevertheless very unsatisfactory. Gale & Church (1994) examine the Add-One case in detail and show that it can give approximately accurate estimates only for data-sets which obey certain quite implausible numerical constraints. Tested on a body of real linguistic data, Add-One gives estimated frequencies which for seen species are always much less accurate even than the maximum likelihood estimator, and are sometimes wrong by a factor of several thousand.

The sole virtue of these techniques is their arithmetical simplicity. But, with a modest increase in complexity of calculation, one can achieve estimators whose performance is far superior. The purpose of this paper is to describe a family of population frequency estimators, the *Good-Turing estimators*, which deserve to be better known to computational linguists than they currently are; and to define a member of this family, the *Simple Good-Turing estimator*, which is easier to understand and to use than the various Good-Turing estimators previously described in the literature, and which is shown to give very satisfactory performance.

2. THE BACKGROUND AND AN EXAMPLE

Good-Turing frequency estimation techniques, the classic exposition of which is Good (1953), emerged from the intellectual partnership between Alan Turing and I.J. Good within the mechanized codebreaking effort at Bletchley Park, Buckinghamshire, during the Second World War; this work depended heavily on inferences about probabilities. (Good-Turing techniques may thus be reckoned as one of the minor fruits of the Bletchley Park enterprise, whose more significant consequences included large shares of responsibility for Allied victory in the war, and for the development of the digital computer. On the general background, see Hodges, 1983; Hinsley & Stripp, 1993.) Good-Turing techniques yield estimates for the population frequencies corresponding to the various observed sample frequencies for seen species, and an estimate for the total population frequency of all unseen species taken together (we shall call this quantity P_0 – note that capital P is used for the sum of the separate probabilities of a number of species, whereas p_i is used for the individual probability of a species i). The techniques do not in themselves tell one how to share

P_0 between the separate unseen species, but this is an important consideration in applying the techniques and we discuss it in §9 below. Also, Good-Turing techniques do not yield estimates for the number of unseen species, where this is not known independently. (Some references on this last issue are Fisher et al., 1943; Goodman, 1949; Good & Toulmin, 1956; McNeil, 1973; Efron & Thisted, 1976.)

In order to introduce Good-Turing concepts, let us take a concrete example, which is drawn from research on speech timing reported in Bachenko & Gale (1993); we shall refer to this as the "prosody example". Assuming a classification of speech segments into consonants, full vowels, and reduced vowels, we wish (for reasons that are not relevant here) to estimate the frequencies in English speech of the various possible sequences containing only the classes "consonant" and "reduced vowel" occurring between two full vowels. That is, the "species" in the population are strings such as VCV, VCRCV, VCCRCRCV, and so on, using C, V, and R to represent the three classes of speech-segment.

Using the TIMIT database as a sample, observed frequencies were extracted for various species; a few examples of the resulting figures are shown in Table 1.

The Appendix shows the complete range of sample frequencies represented in these data, together with the frequencies of the respective sample frequencies; if r is a sample frequency, we write n_r for the number of different species each having that frequency, thus n_r is a "frequency of a frequency". For instance, the third row of the Appendix ($r = 3$, $n_r = 24$) means that there are 24 distinct strings which each occur three times in the data. The sample comprises a total of 30 902 individual strings (this is the sum of the products of the two numbers in each row of the Appendix); that is, $N = 30\,902$. The string VCV, with frequency 7846, is the single commonest *species* in the data. The commonest *frequency* is 1, which is shared by 120 species. As one moves to frequencies greater than 1, the frequencies of the frequencies decline, at first steadily but later more irregularly. These are typical patterns for many kinds of language and speech data.

Table 1. Observed Frequencies of Some Phone-type Strings.

VCV	7846
VCCV	6925
VCCRCRCV	224
VCCRRCCCV	23
VCCRCRV	7
VRCCRCRCV	6
VRCCCRCV	5
VRRCCV	4
VRRCCCV	3
VCCRCRV	2
VRCRCRV	1

3. THE THEORETICAL RATIONALE

We now outline the reasoning underlying the Good-Turing approach to estimating population frequencies from data-sets such as that of the Appendix. The theorems on which the techniques depend are stated without proof; readers wishing to pursue the subject may like to consult Church, Gale, & Kruskal (1991).[3] Some readers may prefer to bypass the present section altogether, in favour of consulting only §6, which presents mechanical "recipe book" instructions for applying the Simple Good-Turing technique without explaining why it works. However, applications of the technique are likely to be more judicious when based on an awareness of its rationale.

We first introduce an additional notation, r^*. Given a particular sample, we write r^* for the estimated number of cases of a species actually observed r times in that sample which *would* have been observed, if the sample were perfectly representative of the population. (This condition would require the possibility of fractional observations.) The quantity r^* will normally be less than r, since if the sample were perfectly representative part of it would be taken up by unseen species, leaving fewer elements of the sample to accommodate the species that actually were observed. Good-Turing techniques consist mainly of a family of methods for estimating r^* (for frequencies $r \geq 1$); given r^*, we es-

timate p_r (which is what we are trying to find) as r^*/N.

Suppose we knew the true population frequencies p_1, p_2,...p_s of the various species. Then we could calculate the *expected frequency* $E(n_r)$ of any sample frequency r; $E(n_r)$ would be $\sum_{i=1}^{s}\binom{N}{r}(p_i)^r(1-p_i)^{N-r}$, where $\binom{N}{r}$ represents the number of distinct ways one can draw r objects from a set of N objects. (That is, the expected frequency of frequency r would be the sum of the probabilities, for each r-sized subset of the sample and each species, that all members of the subset belong to that species and no other sample element belongs to it.) This expectation depends on an idealized assumption that there are no interactions between occurrences of particular species, so that each occurrence of species i is the outcome of something akin to an independent dice-throwing experiment in which one face of the dice represents i and the other faces represent not-i and the probability p_i of getting i rather than not-i is fixed and unchanging: statisticians call this a *binomial* assumption. In reality the assumption is usually false, but often it is false only in ways that have minor, negligible consequences for the overall pattern of occurrences in a sample; in applying statistical methods that incorporate the binomial assumption (including Good-Turing methods) to a particular domain, one must be alive to the issue of whether the binomial assumption is likely to be seriously misleading in that domain. For our example, occurrences of particular strings of consonants and reduced vowels are not truly independent of one another: for some pairs of strings there are several English words which contain both strings at successive points, for other pairs there are no such words. But, within a sizeable database containing many words, these interrelationships are likely to affect the overall pattern of string frequencies sufficiently little to make the binomial assumption harmless.[4]

If we knew the expected frequencies of frequencies, it would be possible to calculate r^*. The central theorem underlying Good-Turing methods states that, for any frequency $r \geq 1$:

$$(1) \qquad r^* = (r+1)\frac{E(n_{r+1})}{E(n_r)}$$

A corollary states that:

$$(2) \qquad P_0 = \frac{E(n_1)}{N}$$

In reality we cannot calculate exact figures for expected frequencies of frequencies, because they depend on the probabilities of the various species, which is what we are trying to find out. However, we have figures for the observed frequencies of frequencies, and from these we can infer approximations to the expected frequencies.

Take first equation (2). This involves only the expected frequency of sample frequency 1. In the sort of data we are considering, where there are few common species but many rare species, frequency 1 will always be the commonest sample frequency, and the actual figure for n_1 is likely to be a close approximation to $E(n_1)$ – compare the fact that the oftener one tosses a coin, the surer one can be that the cumulative proportion of heads will be close to one half. Thus it is reasonable to estimate P_0 as equal to n_1/N. In the example, n_1 is 120, hence our estimate of the total probability of all unseen species of strings is 120/30902, or 0.0039. If another ten thousand strings were sampled from speech comparable to that sampled in the TIMIT database, we estimate that 39 of them would represent some string or strings not found in TIMIT.

As we move to higher sample frequencies, the data become increasingly "noisy": already at $r = 5$ and $r = 7$ in the Appendix we see cases where n_r is greater than n_{r-1}, although the overall trend is for n_r to decrease as r increases. Furthermore there are many gaps in the list of observed sample frequencies; thus for our example one could not get a sensible r^* figure in the case of $r = 10$ by substituting actual for expected frequencies of frequencies in equation (1), because the frequency $r + 1$, i.e. 11, does not occur at all (n_{11} is zero, so 10* calculated in this way would also be zero, which is absurd). As one moves towards higher values of r, the gaps where $n_r = 0$ become larger. What we need

is a technique for smoothing the irregular and "gappy" series of n_r figures into a regular and continuous series, which can be used as good proxies for the unknowable $E(n_r)$ figures in equation (1).

Much of Good's 1953 paper concerned alternative techniques for smoothing observed series of frequencies of frequencies. The reason for speaking of Good-Turing *techniques*, in the plural, is that any concrete application of the above concepts requires a choice of some particular method of smoothing the n_r figures; not all methods will give equally accurate population-frequency estimates in a given domain. Some techniques (including the smoothing technique of Church & Gale, 1991) are mathematically quite elaborate. The Simple Good-Turing method is relatively easy to use, yet we shall show that it gives good results in a variety of tests.

4. LINEAR SMOOTHING

To gain an intuitive grasp of SGT smoothing, it is helpful to visualize the data graphically. Fig. 1 plots n_r against r for our example. Because the ranges of values for both r and n_r include values clustered close together in the lower reaches of the respective ranges and values separated widely in the upper reaches (as is typical of linguistic data), the plot uses a logarithmic scale for both axes.

For lower sample frequencies the data points group round a northwest-to-southeast trend, but at higher sample frequencies the trend becomes horizontal along the line $n_r = 1$. This angular discontinuity in Fig. 1 does not correspond to any inherent property of the population. It is merely a consequence of the finite size of the sample: a sample frequency may occur once or not at all but cannot occur a fractional number of times. When using observed frequencies of frequencies to estimate expected frequencies of frequencies for high sample frequencies, we ought to take account not only of the fact that certain high r values correspond to positive n_r values but also of the fact that neighbouring r values correspond to zero n_r values. Following

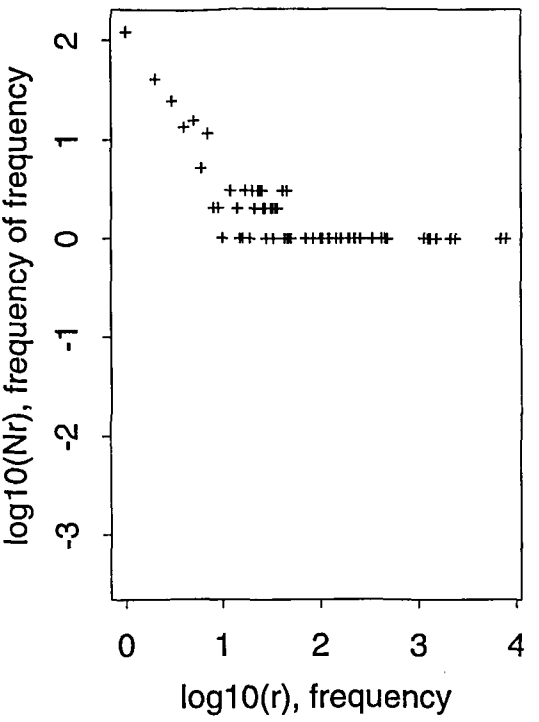

Fig. 1.

Church & Gale (1991), we do this by averaging positive n_r values with the surrounding zero values. That is, we define a new variable Z_r as follows: for any sample frequency r, let r' be the nearest lower sample frequency and r'' the nearest higher sample frequency such that $n_{r'}$ and $n_{r''}$ are both positive rather than zero. Then $Z_r = 2n_r/(r'' - r')$. For low r, r' and r'' will be immediately adjacent to r, so that $r'' - r'$ will be 2 and Z_r will be the same as n_r; for high r, Z_r will be a fraction, sometimes a small fraction, of n_r.[5] Most of our estimates of expected frequencies will be based on Z_r rather than directly on n_r.

Fig. 2a plots Z_r against r for our sample on the same log-log scales as in Fig. 1. The discontinuity of Fig. 1 has disappeared in Fig. 2a: the data points all group fairly well along a single common trend. Furthermore, not only does Fig. 2a display a homogeneous trend, but this trend is a straight line. That is not surprising: G.K. Zipf argued that distribution patterns for many linguistic and other behavioural elements are

approximately log-linear.[6] We have examined perhaps a dozen radically different language and speech data-sets, and in each case on a log-log plot the points group round a straight line (with a slope between –1 and –2). The Simple Good-Turing technique exploits the fact that such plots typically show linear trends.

Any method of smoothing data must, if it is to be usable for our present purpose, satisfy certain prior expectations about r^*. First, we expect r^* to be less than r, for all nonzero values of r; secondly, we expect r^*/r to approach unity as r increases. The first expectation follows from the fact that observed sample frequencies must be reduced in order to release a proportion of sample elements to accommodate unseen species. The second expectation reflects the fact that the larger r is, the better it is measured, so we want to take away less and less probability as r increases.

It is not at all easy to find a method for smoothing Z_r figures that will ensure the satisfaction of these prior expectations about r^*. However, a downward-sloping log-log line is guaranteed to satisfy them. Since a straight line is also the simplest possible smooth, part of the SGT technique consists of using the line of best fit to the (log r, log Z_r) points to give our proxy for $E(n_r)$ values when using equation (1) to calculate r^*. We shall write $S(r)$ ("smoothed Z_r") for the value into which this line takes a sample frequency r.[7] Fig. 2b shows the line of best fit superimposed on the data points of Fig. 2a.

But, for the lowest few values of r, observed n_r values may well be more accurate than any smoothed values as estimates of $E(n_r)$. Therefore the other aspect of the SGT technique consists of a rule for switching between n_r and $S(r)$ as proxies for $E(n_r)$ when calculating r^* – for switching between *raw* and *smoothed proxies*, we shall say. The rule is that r^* is calculated using n_r rather than $S(r)$ as proxy for $E(n_r)$ for r from 1 upwards so long as these alternative methods of calculating r^* give significantly dif-

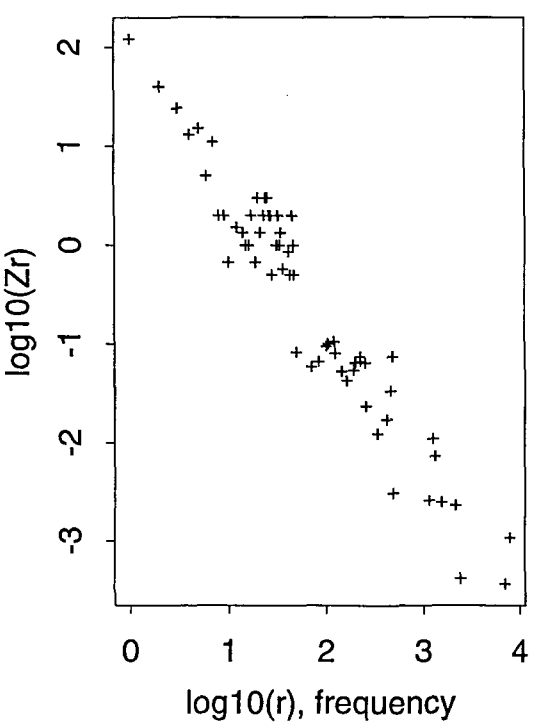

Fig. 2a.

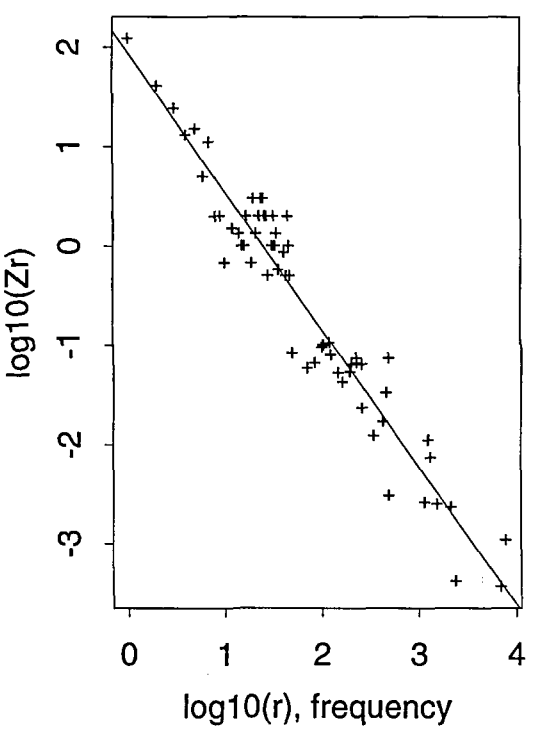

Fig. 2b.

ferent results. (The general pattern is that, as r increases from 1, there will be a short stretch of values for which the alternative r^* estimates are significantly different, then a short stretch of values where the pairs of r^* estimates oscillate between being and not being significantly different, and then above a certain value of r the pairs of estimates will never be significantly different.) Once the lowest value of r is reached for which n_r and $S(r)$ give estimates of r^* which are not significantly different, $S(r)$ is used to calculate r^* for that value and for all higher values of r.

Pairs of r^* estimates may be considered significantly different if their difference exceeds 1.96 times the standard deviation (square root of variance) of the estimate based on n_r (since, assuming a Gaussian distribution of that estimate, the probability of such a difference occurring by chance is less than the accepted .05 significance criterion).[8] The variance in question is approximately equal to

$$(r+1)^2 \frac{n_{r+1}}{n_r^2}\left(1+\frac{n_{r+1}}{n_r}\right).$$[9]

It is the adoption of a rule for switching between smoothed and raw frequencies of frequencies as proxies for expected frequencies of frequencies which allows the SGT method to use such a simple smoothing technique. Good-Turing methods described previously have relied on smoothed proxies for all values of r, and this has forced them to use smoothing calculations which are far more daunting than that of SGT.[10]

One further step is needed before the SGT estimator is completely defined. Because it uses proxies for the true expected frequencies of frequencies $E(n_r)$, we cannot expect the estimated probabilities yielded by the SGT technique to sum to one, as they should. Therefore each estimated probability generated as discussed above has to be *renormalized* by dividing it by the total of the unnormalized estimates and multiplying by the estimated total probability of seen species, $1 - P_0$.

Applying the technique defined above to the prosody data in the Appendix gives a line of best fit $\log S(r) = -1.389 \log r + 1.941$ (with $S(r)$ interpreted as discussed above). For comparison with Table 1, we show in Table 2 the r^* and p_r figures estimated by SGT for the same selection of species. In this particular example, as it happens, even for $r = 1$ the alternative calculations of r^* give figures that are not significantly different, so values based on smoothed proxies are used throughout; but that is a chance feature of this particular data-set, and in other cases the switching rule calculates r^* from raw proxies for several values of r – for instance, in the "Chinese plurals" example of the following section, raw proxies are used for $r = 1$ and $r = 2$.

Table 2. Estimated Population Frequencies of Some Phone-type Strings.

	r	r^*	p_r
VCV	7846	7839.	0.2537
VCCV	6925	6919.	0.2239
VCCRCRCV	224	223.4	0.007230
VCCRRCCCV	23	22.60	0.0007314
VCCRCRV	7	6.640	0.0002149
VRCCRCRCV	6	5.646	0.0001827
VRCCCRCV	5	4.653	0.0001506
VRRCCV	4	3.664	0.0001186
VRRCCCV	3	2.680	8.672e-05
VCCRCRV	2	1.706	5.522e-05
VRCRCRV	1	0.7628	2.468e-05

5. OPEN V. CLOSED CLASSES: CHINESE PLURALS

A second example, illustrating an additional use of the concepts under discussion, is taken from the field of Chinese morphology. Chinese has various devices to mark the logical category of plurality, but (unlike in European languages) this category is by no means always marked in Chinese. For instance, there is a plural suffix *men* which can be added to personal pronouns and to some nouns; but many nouns never take *men* irrespective of whether they are used with plural reference in a particular context, and nouns which can take *men* will not always do so even when used with plural reference.

In connexion with work reported in Sproat et al. (1994) on the problem of automatically segmenting written Chinese into words, it was de-

sirable to establish whether the class of Chinese nouns capable of taking *men* is open or closed.[11] Dictionaries are silent on this point and grammatical descriptions of the language tend to be less than wholly explicit; but it is important for word segmentation – if the class of nouns that can take *men* is closed, an efficient algorithm could list them, but if the class is open some other technique must be deployed.

The frequencies of various nouns in *men* found in a (manually segmented) corpus of Chinese were tabulated, the commonest case being *rén-men* "people" which occurred 1918 times. Altogether there were 6551 tokens exemplifying 683 types of *men* plural. Some sample r, n_r figures are shown in Table 3.

The question whether a linguistic class is open or closed is not the same as the question whether the number s of species in a population is finite or infinite. Asking whether a large class of linguistic items should be regarded as mathematically infinite tends to be a sterile, philosophical question. The number of words in the English vocabulary, for instance, must arguably be finite: for one thing because only a finite number of users of the language have lived, each producing a finite number of word-tokens in his lifetime, and word-types must be fewer than word-tokens; for another thing, because any English word is a string of tokens of a few dozen character-types, and it is probably safe to say that a word more than twice as long as the longest that has occurred would be unusable. But if the English vocabulary is finite, it is certainly an open class: for practical purposes it "might as well" be infinitely large. The question whether a class is closed or open in this sense might be glossed as whether a sample of a size that is practical to assemble will contain examples of a large fraction, or only a small fraction, of all the species constituting the class. A corpus of tens of millions of English word-tokens will exemplify only a tiny fraction of all the word-types used in the English language.

Table 3. Some Frequencies of Frequencies for Chinese-plural Data.

r	n_r
1	268
2	112
3	70
4	41
5	24
6	14
7	15
400	1
1918	1

In terms of the statistical concepts under discussion, if a class is closed we expect to find $1^* > 1$. With a closed class one will soon see most of the species, so the number of species seen just once will tend to become small. For the Chinese plurals data, $1^* = 2n_2/n_1 = 2 \times 112/268 = 0.84$, which is convincingly less than unity;[12] so we conclude that the class of Chinese nouns forming a plural in *men* is open, at least in the sense that it must be very much larger than the 683 observed cases. This harmonizes with the statement in Y.R. Chao's authoritative grammar of Chinese (Chao, 1968: 244-245) according to which *men* can be suffixed to "words for persons" (and, in certain regional dialects, to some other nouns), which suggests that *men* plurals form an open class.

Rather than giving a series of figures analogous to Table 2 for the Chinese plurals example, a clearer way of showing the reader the nature of a set of SGT estimates is via a plot of r^*/r against r – such a plot is enlightening whenever Good-Turing techniques are applied. Fig. 3 plots r^*/r against r for both the prosody and the Chinese-plural examples, representing the two sets of data points by 'p' and 'c' respectively.

The Chinese plurals example needs more probability set aside for unseen types than does the prosody example (.04 versus .004); but it has twice as many types and five times as many tokens to take this probability from, so the ratios of r^* to r are not so very different between the two cases. The fact that 1^* and 2^* in the Chinese plurals case are based on raw proxies which yield estimates that are significantly larger than the alternative estimates based on smoothed proxies – as is apparent from the distribution of 'c' symbols in Fig. 3 – hints that the class may not be entirely open-ended, but if required to

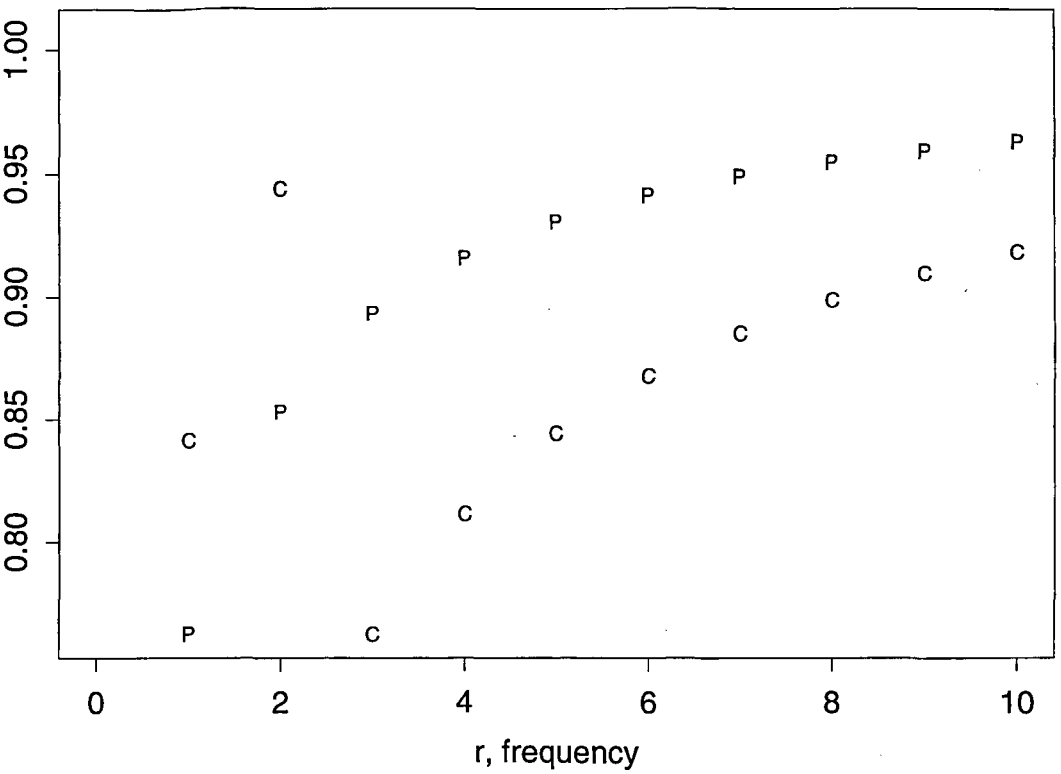

Fig. 3.

categorize it on one or the other side of what is in reality a continuum between open and closed classes, on the basis of the data used here we would probably do well to treat it as open.

6. THE PROCEDURE STEP BY STEP

This section presents a complete but totally mechanical statement of the SGT algorithm. No rationale is offered in this section. §3 covered the reasons for the steps in the algorithm which we now present.[13]

Our data are a sample of individuals belonging to various species. On the basis of the numerical properties of the sample we shall assign values to an integer variable N and real variables P_0, N', a, b, and to the cells of a table. The table is to have as many rows as there are distinct species frequencies in the data, and seven columns labelled r, n, Z, $\log r$, $\log Z$, r^*, p. The values in the r and n columns will be integers, those in the other columns will be reals (in a concrete computer realization of the algorithm it may be convenient to use separate arrays).

First, tabulate the various species frequencies found in the sample, and the numbers of distinct species occurring with each species frequency, in the r and n columns respectively. For instance, a row with $r = 3$ and $n = 24$ will mean that there are 24 different species each represented in the sample by three individuals. Enter these pairs of numbers in the appropriate columns in such a way that r values always increase between successive rows: the first row will have $r = 1$, and the last row will have the frequency of the commonest species in the r column. It is convenient not to include rows in the table for frequencies that are not represented in the sample: thus the n column will contain no zeros, and many integers between 1 and the highest species frequency will appear nowhere

in the r column. Thus, for the prosody example of §2 these first two columns will look like the Appendix.

We shall use the values in the r column to identify the rows, and they will appear as subscripts to the labels of the other columns to identify cell values. For instance Z_i will mean the contents of the cell in the Z column and the row which has i in the r column (not the i'th row).

Assign to N the sum of the products of the pairs of integers in the r and n columns. This will be the number of individuals in the sample. (In practice the value of N will often have been ascertained at an earlier stage, but if not it can be done in this way.)

Assign to P_0 the value n_1/N (where n_1 represents the value in the n column and the row for which $r = 1$). P_0 is our estimate of the total probability of all unseen species. If the identity of the various unseen species is known, P_0 should be divided between them by reference to whatever features of the species may suggest prior probabilities for them (cf. §9).

Enter values in the Z column as follows. For each row j, let i and k be the values in the r column for the immediately previous and immediately following rows respectively (so that $k > i$). If j is the first row, let i be 0; if j is the last row, let k be $2j - i$. Set Z_j to the value $2n_j/(k-i)$.

Enter the logarithms of the r and Z values in the corresponding rows of the log r and log Z columns. Use regression analysis to find the line of best fit $a + b \log r$ to the pairs of values in the log r and log Z columns. (Regression analysis to find the "line of best fit" or "line of least squares" for a set of data points is a simple and standard manipulation described in most elementary statistics textbooks; see e.g. Press et al., 1988: 523-526, which includes computer coding.[14])

We shall use "$S(r)$" as an abbreviation for the function antilog $(a = b \log r)$. (If base-10 logarithms are used, antilog(x) means 10^x.) Working through the rows of the array in order beginning with the row $r = 1$, begin by calculating for each value of r the two values x and y defined by equations (3) and (4) below. If inequality (5) holds, then insert x in the r^* column. ($|x - y|$ means the absolute difference between x and y.) If (5) does not hold, insert y in the r^* column, and cease to calculate x values: for all subsequent rows insert the respective y value in the r^* column.

$$(3) \qquad x = (r+1)\frac{n_{r+1}}{n_r}$$

$$(4) \qquad y = (r+1)\frac{S(r+1)}{S(r)}$$

$$(5) \qquad |x-y| > 1.96 \times \sqrt{(r+1)^2 \frac{n_{r+1}}{n_r^2}\left(1+\frac{n_{r+1}}{n_r}\right)}$$

(Since the values in the r column are not continuous, in theory the instruction of the preceding paragraph might be impossible to execute because the calculation of x could call for an n_{r+1} value when the table contained no row with the corresponding value in the r column. In practice this is likely never to happen, because the switch to using y values will occur before gaps appear in the series of r values. If it did ever happen, the switch to using y values would have to occur at that point.)

Let N' be the total of the products $n_r r^*$ for the various rows of the table. For each row calculate the value $(1-P_0)\frac{r^*}{N'}$ and insert it in the p column.

Each value p_r in this column is now the SGT estimate for the population frequency of a species whose frequency in the sample is r.

7. TESTS OF ACCURACY: A MONTE CARLO STUDY

SGT gives us estimates for species probabilities in the prosody example, but although we have theoretical reasons for believing the estimates to be good we have no way of determining the true probabilities for this example, and hence no objective way of assessing the accuracy of the method. We now present two cases where we do know the answers.

The first is a *Monte Carlo* study, meaning that data-sets are created artificially, using a

(pseudo-)random number generator, in order to constitute samples from populations with known statistical properties: statistical inference techniques can be applied to such samples and their findings compared with the properties which the population is known to possess. Such techniques are well established in statistical research.

For this study, we constructed a set of sample texts each containing 100 000 word-tokens. Each text was constructed by drawing tokens randomly from an ordered list $w_1, w_2, \ldots, w_s$ of word-types, with the probability of drawing a token of the i'th type being made proportional to i^z for some z less than -1. Specifically, for a text with given s and z the probability of w_i ($i \leq s$) was $\frac{i^z}{\sum_{j=1}^{s} j^z}$. Such a distribution is called a Zipfian distribution with exponent z (the reference here being to "Zipf's Law" – cf. note 6 above).

The study used five values of s (vocabulary size), namely 5000, 10 000, 25 000, 50 000, and 100 000, and four values of z, namely -1.1, -1.2, -1.3, -1.4. One text was constructed for each combination of vocabulary size and exponent, giving twenty texts in all. At most 15 000 word-types were represented in any one text, thus the spectrum of vocabulary sizes extended from cases where the finite nature of the vocabulary was significant to cases where it is impossible to tell from a 100 000-token text whether the vocabulary is finite or infinite. The range of exponents are representative of values seen in real linguistic data.

The question to which Good-Turing methods estimate the answer, for any one of these texts, is what the average probability is of those types which are represented exactly r times in the text (for any integer r). Each individual type is assigned a specific probability by the model used to construct the text, therefore the true average probability of types which are represented by r tokens can easily be calculated. Since the most difficult cases are those where r is small, we assessed accuracy over the range $1 \leq r \leq 10$. Average probabilities for r in this range have two to three significant figures.

We compared the performance of the Simple Good-Turing method on these data with the performance of three other frequency-estimation techniques which a researcher might be inclined to use: two variants of the additive method of §1, and the *Deleted Estimate* of Jelinek & Mercer (1985).[15]

Fienberg & Holland (1972) survey six variants of the additive method, which all share the advantage of giving nonzero estimates for unseen species frequencies but differ with respect to choice of the figure k which is added to observed sample frequencies. They discuss three "*a priori* values": 1 (as in our §1), $\frac{1}{2}$, and $\frac{1}{s}$ (where s is the number of species, so that one observation is added to the total number of observations to renormalize – this choice of k was advocated by Perks, 1947: 308); and three "empirical values", meaning that k is determined in different ways by the properties of the particular set of observations under analysis. For the kind of data Fienberg & Holland discuss, they suggest that 1 is too large a value for k and $\frac{1}{s}$ too small, but that all four other choices are reasonable. We have chosen to assess the additive method here using $k = \frac{1}{2}$ and $k = \frac{1}{s}$ as two representative choices (we refer to the additive estimator using these values for k as *Add-Half* and *Add-Tiny* respectively): Add-Half is very similar to Add-One but has somewhat greater theoretical justification,[16] and Add-Tiny has the superficial attraction of minimally distorting the true observations. We do not separately assess Add-One, because it is so similar to Add-Half, and we do not assess Fienberg & Holland's "empirical" estimators because language and speech researchers attracted to the simplicity of the additive method would scarcely be tempted to choose these variants.

Add-Half and Add-Tiny are extremely simple to apply, and they may be useful for "quick and dirty" preliminary studies. But we shall see that they both perform too poorly on the Monte Carlo data-sets to seem worth considering for serious investigations.[17]

A theoretically respectable alternative to Good-Turing methods is the well established statistical technique of *cross validation*. It has

been applied to linguistic data under the name "Deleted Estimate" by Jelinek & Mercer (1985), and see also Nádas (1985). Cross validation requires calculations which are more demanding than those of SGT, but they are by no means beyond the resources of modern computing facilities. For present purposes we examine the simplest case, two-way cross validation ("2CV").

The Good-Turing estimator is based on a theorem about the frequency one would expect in a hypothetical additional sample for species occurring with a given frequency r in an observed sample. Jelinek & Mercer begin by defining a *held-out* estimator which turns this concept from hypothesis to reality, creating an actual additional sample by dividing an available text sample into two halves, called *retained* and *held-out*, corresponding respectively to the actual sample and the hypothetical additional sample of the Good-Turing approach. Let n_r be the number of species which are each represented r times in the retained subsample, and let C_r be the total number of occurrences of those particular species in the held-out subsample. Then C_r/n_r is used as the adjusted frequency r^* from which the estimated population frequency is derived.

As it stands this technique is inefficient in the way it extracts information from available data. Two-way cross validation (such as Jelinek & Mercer's Deleted Estimate) uses the data less wastefully; it combines two held-out estimates made by swapping the roles of held-out and retained subsamples. If we denote the two halves of the data by 0 and 1, we write n_r^0 for the number of species each occurring r times in subsample 0, and C_r^{01} for the total number of occurrences in subsample 1 of those particular species; n_r^1 and C_r^{10} are defined correspondingly. The two held-out estimators would be C_r^{01}/n_r^0 and C_r^{10}/n_r^1; the Deleted Estimate combines the underlying measurements by using equation (6) to estimate r^*:

$$r^* = \frac{C_r^{01} + C_r^{10}}{n_r^0 + n_r^1} \tag{6}$$

Cross validation does not make the binomial assumption made by Good-Turing methods; it makes only the much weaker assumptions that the two subsamples are generated by statistically identical processes, and that the probability of a species seen r times in a sample of size N is half that of a sample seen r times in a sample of size $N/2$. Cross validation need not be "two-way"; available data may be divided into three or more subsamples. However, even two-way cross validation is a computationally intensive procedure, and the computational demands grow as the number of subsamples is increased.

One consideration distinguishing the additive techniques from both the Good-Turing and cross validation approaches is that the former, but not the latter, require knowledge of the number of unseen species. In a real-life application where the "species" are vocabulary items this knowledge would not ordinarily be available. Nevertheless, we have allowed ourselves to use it, in order to produce results from the additive techniques for comparison with the SGT and 2CV results. Since both the additive techniques used prove inferior to both SGT and 2CV, allowing the former to use the extra information has not distorted the validity of our overall conclusions.

Because true and estimated probabilities can vary by several orders of magnitude, it is convenient to express the error in an estimated probability as the logarithm of its ratio to the true probability. For each of the four estimation methods, Table 4 gives the root mean square of the base-10 logarithms of these ratios for 11 values of r from 0 to 10 for each of the twenty datasets (five values of s times four values of z). We shall refer to the root mean square of the logarithms of a set of estimated-probability/true-probability ratios as the *average error* of the set.

Table 4. Overall Average Error for Four Estimators.

Method	*RMS error*
Add-Half	0.47
Add-Tiny	2.62
SGT	0.062
2CV	0.18

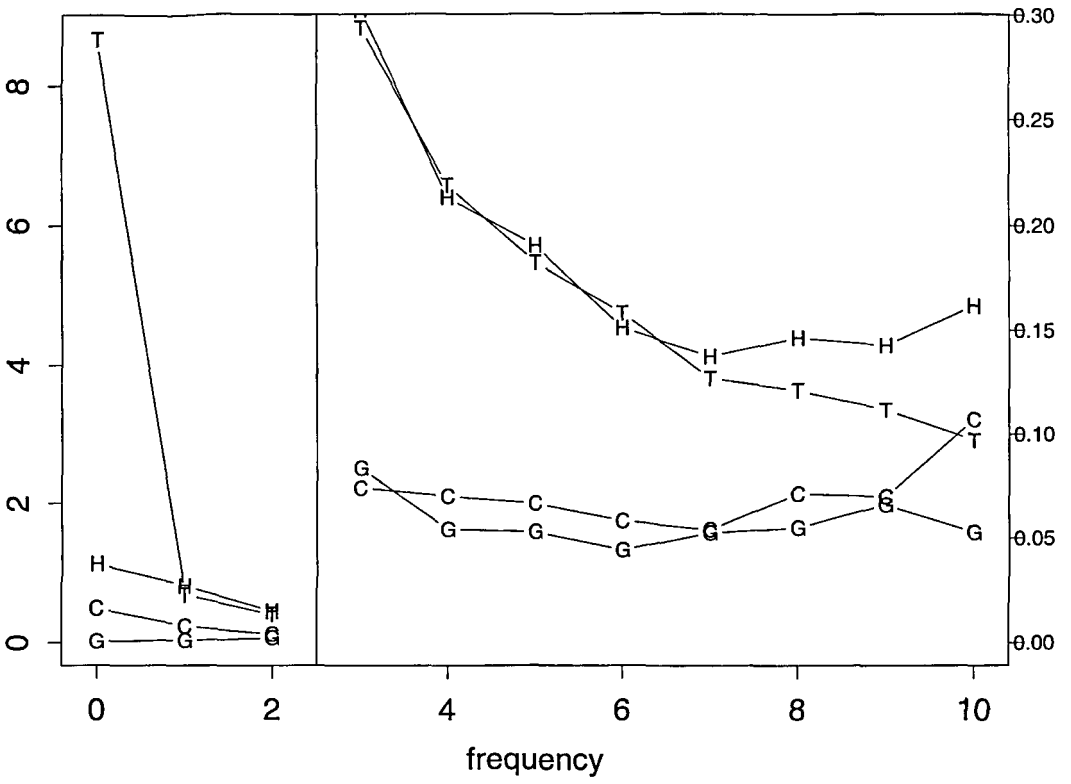

Fig. 4.

Add-Half gets the order of magnitude correct on average, but Add-Tiny fails even to achieve that on the Monte Carlo data. Of the four methods, SGT gives the best overall results.

Breaking down the overall error rates by different values of r shows where the different techniques fail. In Fig. 4, different plotting symbols represent the different methods as follows:

H Add-Half
T Add-Tiny
G Simple Good-Turing
C Two-Way Cross Validation

In order to accommodate the full frequency range in a single Figure, Fig. 4 uses two scales for average error: the scale on the left applies for $r \leq 2$, the scale on the right applies for $r > 2$. Each point represents the average error for the twenty combinations of vocabulary size and exponent. We see that the additive methods are grossly wrong for unseen species, and remain less accurate than SGT and 2CV over the range of positive frequencies shown.

By eliminating the data points for the additive methods, Fig. 5 is able to use a less compressed vertical scale to display the error figures for the SGT and 2CV methods. We see that for r greater than about two, the performance of SGT is comparable to that of 2CV, but that the latter is poor for $r \leq 2$. (It is of course possible that multi-way cross validation would give better performance for small r; we do not know whether that is so or not, but we have seen that multi-way cross validation is far more demanding computationally than SGT.)

The performance of SGT in particular is displayed in more detail in Fig. 6, which further expands the vertical scale. We see that SGT does best for small r and settles down to an error of about 5% for large r. There is an intermediate zone of a few frequencies where SGT does less well. This is because the SGT method

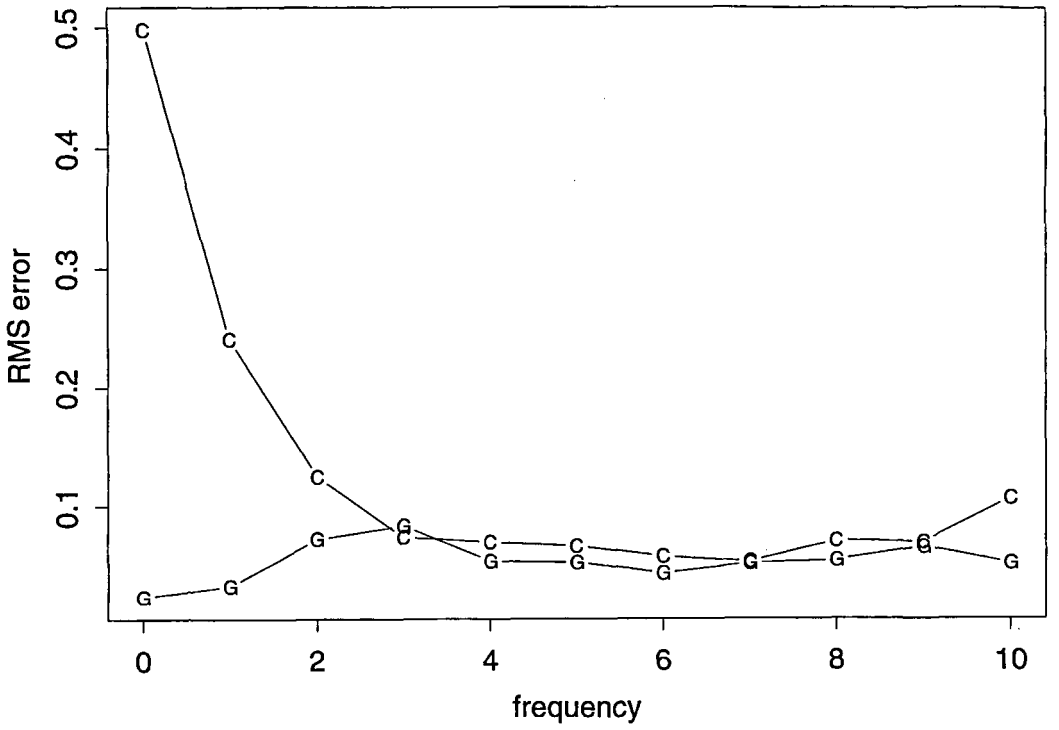

Fig. 5.

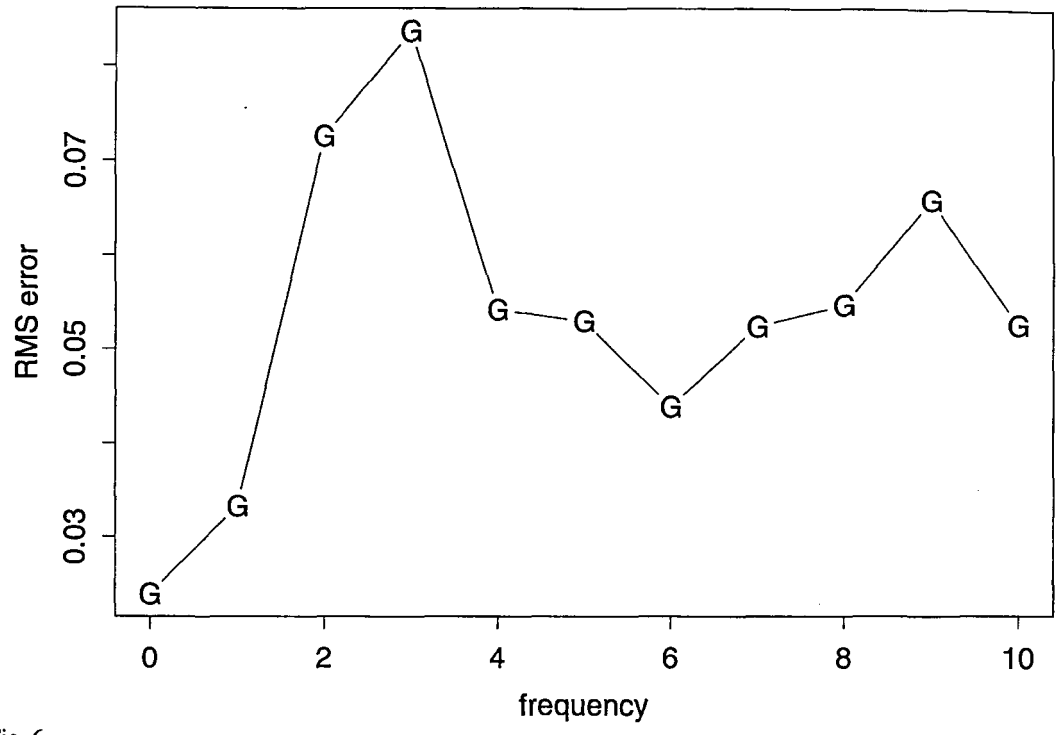

Fig. 6.

switches between estimates based on raw proxies for small r, and estimates based on smoothed proxies for higher r: in the switching region both of these estimation methods have problems.

Figs. 7 and 8 show how average error in the SGT estimates varies with vocabulary size and with Zipfian exponent respectively. We see that there is no correlation between error level and vocabulary size, and little evidence for a systematic correlation between error level and exponent (though the average error for the largest exponent is notably greater than for the other three values). Furthermore the range over which average error varies is much smaller for varying exponent or (especially) varying vocabulary size than for varying r.

Error figures obtained using real linguistic data would probably be larger than the figures obtained in this Monte Carlo study, because word-tokens are not binomially distributed in real life.

8. TESTS OF ACCURACY: A BIGRAM STUDY

A second test of accuracy is based on the findings reported in Tables 1 and 2 of Church & Gale (1991). This study used a 44-million-word sample of English comprising most of the different articles distributed by the Associated Press newswire in 1988 (some portions of the year were missing, and the material had been processed in order to eliminate identical or near-identical articles). Each bigram in the sample was assigned randomly to one of two subsamples: thus, although we may not know how representative 1988 AP newswire stories are of wider linguistic populations such as modern journalistic American English, what matters for present purposes is that we can be sure the two subsamples come as close as possible to both representing exactly the same population (namely, 1988 AP newswire English).

Since Good-Turing techniques predict fre-

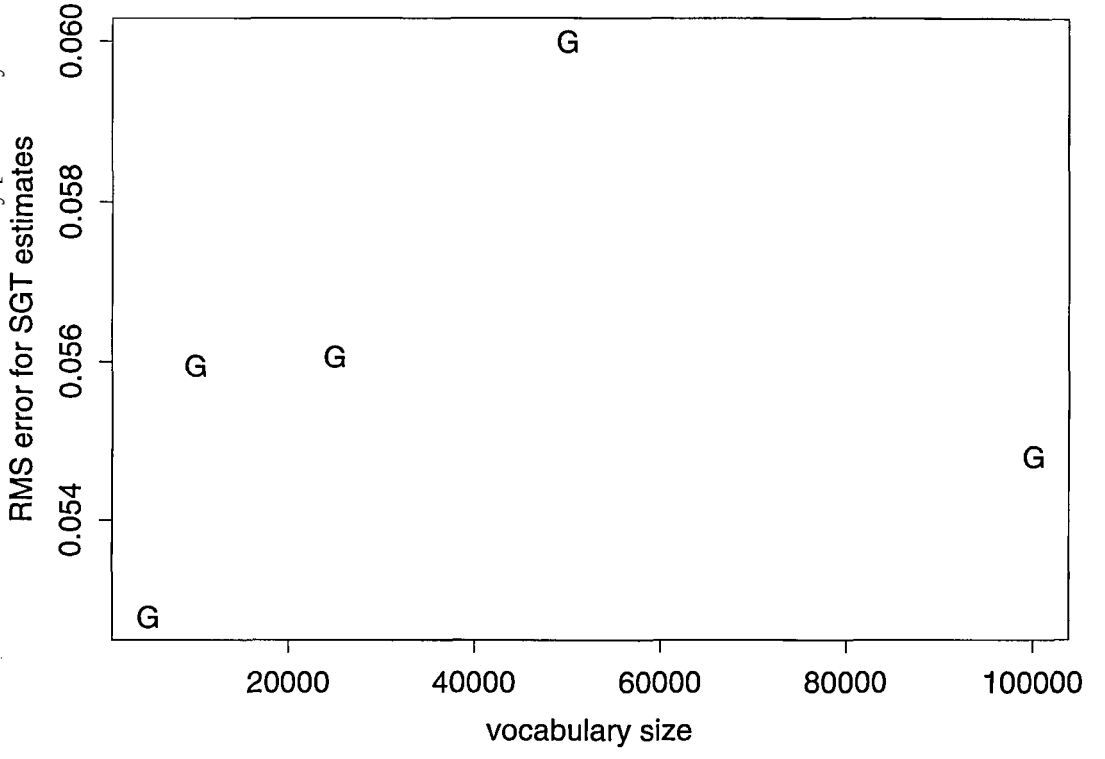

Fig. 7.

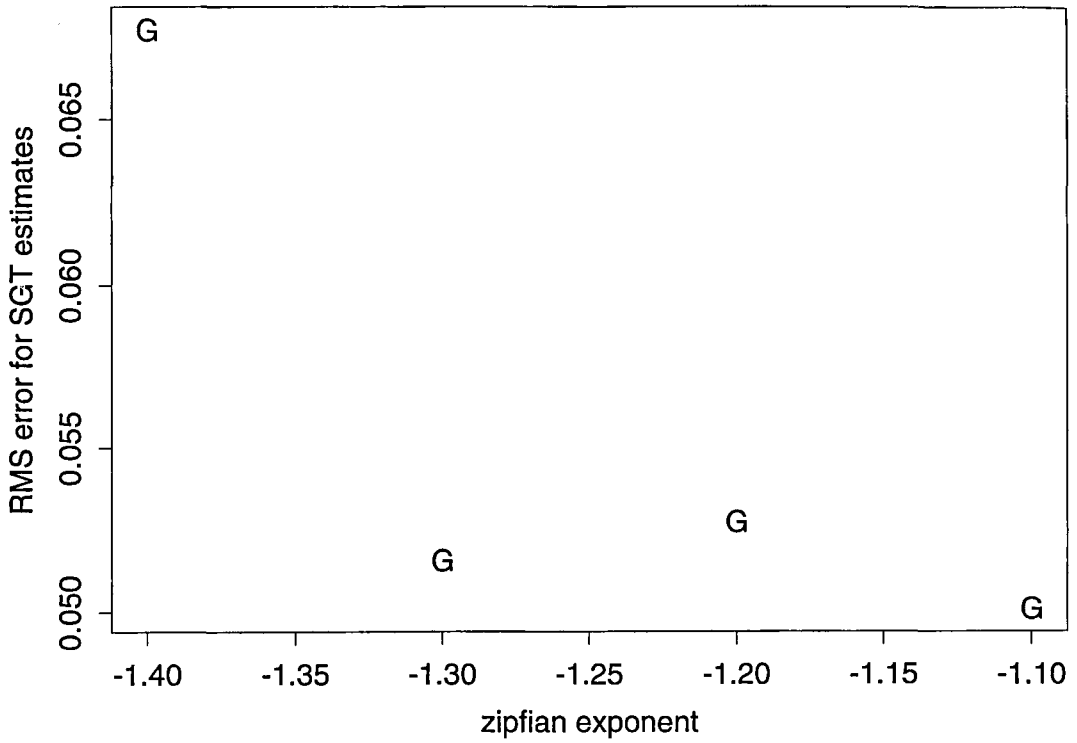

Fig. 8.

quencies within a hypothetical additional sample from the same population as the data, whereas the "held-out" estimator of §7 reflects frequencies found in a real additional sample, we can treat the held-out estimator based on the two 22-million-word AP subsamples as a standard against which to measure the performance of the SGT estimator based on just the "retained" subsample. Table 5 compares the r^* estimates produced by held-out and SGT methods for frequencies from 1 to 9.

In this example, the huge values for n_r meant that for all r values shown the SGT method selected the estimate based on raw rather than smoothed proxies. In no case does the SGT estimate deviate by more than 1% from the held-out estimate. The quantity of data used makes this an untypical example, but the satisfactory performance of the SGT technique is nevertheless somewhat reassuring. (The largest error in the estimates based on smoothed proxies is 6%, for $r = 0$ – that is, for the P_0 estimate.)

Table 5. Alternative r^* Estimates for the Bigram Data.

r	n_r	r^*_{SGT}	r^*_{HO}
1	2 018 046	0.446	0.448
2	449 721	1.26	1.25
3	188 933	2.24	2.24
4	105 668	3.24	3.23
5	68 379	4.22	4.21
6	48 190	5.19	5.23
7	35 709	6.21	6.21
8	37 710	7.24	7.21
9	22 280	8.25	8.26

9. ON ESTIMATING THE PROBABILITIES OF UNSEEN SPECIES

Good-Turing techniques give an overall estimate P_0 for the probability of all unseen species taken together, but in themselves they can give no guide to the individual probabilities of the separate unseen species.

Provided the number of unseen species is known, the obvious approach is to divide P_0 equally between the species. But this is a very unsatisfactory technique. Commonly, the "species" in a linguistic application will have internal structure of some kind, enabling shares of P_0 to be assigned to the various species by reference to the probabilities of their structural components: the resulting estimates may be rather inaccurate, if probabilities of different components are in reality not independent of one another, but they are likely to be much better than merely assigning equal probabilities to all unseen species.

The bigram study discussed in §8 offers one illustration. Many bigrams will fail to occur in even a large language sample, but the sample gives us estimated unigram probabilities for all word-types it contains (that is, probabilities that a word-token chosen at random from the population represents the respective word-type). Writing $p(w)$ for the estimated unigram probability of a word w, the bigram probability of any unseen bigram w_1w_2 can be estimated by taking the product $p(w_1)p(w_2)$ of the unigram probabilities and multiplying it by P_0/P_0' (where P_0 is the Good-Turing estimate of total unseen-species probability as before, and P_0' is the sum of the products $p(w_i)p(w_j)$ for all unseen bigrams w_iw_j – multiplying by P_0/P_0' is a renormalization step necessary in order to ensure that estimated probabilities for all seen and unseen bigrams total unity). This technique is likely to overestimate probabilities for unseen bigrams consisting of two common words which would usually not occur together for grammatical reasons (say, *the if*), and to underestimate probabilities for two-word set phrases that happen not to occur in the data, but over the entire range of unseen bigrams it should perform far better on average than simply sharing P_0 equally between the various cases.

A second example is drawn from the field of syntax. The SUSANNE Corpus (Sampson, 1995) is a 130 000-word subset of the Brown Corpus of written American English, grammatically analysed according to a rigorously-specified analytic scheme.[18] Syntactic constituents are classified in terms of a fine-grained system of grammatical properties, which in some cases allows for many more distinct constituent-types than are represented in the Corpus.

The following figures relate to a version of the SUSANNE Corpus from which 2% of paragraphs chosen at random had been excluded in order to serve as test data for a research project not relevant to our present concerns: thus the sample studied comprises about 127 000 words. Taking the syntactic category "noun phrase" for investigation, the sample contains 34 204 instances, classified by the SUSANNE annotation scheme into 74 species. For instance, there are 14 527 instances of Ns, "noun phrase marked as *singular*", which is the commonest species of noun phrase; there are 41 instances of Np@, "appositional noun phrase marked as plural"; one of the species represented by a single instance is Nj", "noun phrase with adjective head used vocatively". However, the number of possible species implied by the annotation scheme is much larger than 74. A noun phrase may be proper or common, it may be marked as singular, marked as plural, or unmarked for number, and so on for six parameters having between two and six values, so that the total number of species is 1944.

The number of species seen once, n_1, is 12; therefore the Good-Turing estimate of P_0, the total probability of unseen species, is 12/34204 = 0.00035. (The SGT estimate for P_1 is 2.6e–05.) Since each noun-phrase species is defined as a conjunction of values on six parameters, counts of the relative frequencies of the various values on each parameter can be used to estimate probabilities for unseen species. For instance, on the number-marking parameter, to two significant figures 0.64 of noun phrases in the sample are marked as singular, 0.22 are marked as plural, and 0.13 are unmarked for number. If one estimates the probability for each unseen species by multiplying together these probabilities for the various parameter-values which jointly constitute the species, then P_0', the sum of the products for the 1870 unseen species, is 0.085; some samples of estimated probabilities for individual unseen species are shown in Table 6.

In Table 6, Nas+ represents a conjunct intro-

Table 6. Estimated Probabilities for some Unseen Noun Phrase Types.

Nas+	$.0013 \times P_0/P'_0$	=	5.4e–06
Nyn	$.00015 \times P_0/P'_0$	=	6.2e–07
Njp!	$2.7e{-}07 \times P_0/P'_0$	=	1.1e–09

duced by a co-ordinating conjunction and marked as subject and as singular, for instance the italicized phrase in a hypothetical sequence "my son *and he* were room-mates". In English it is more usual to place the pronoun first in such a co-ordination; but intuitively the quoted phrase seems unremarkable, and the calculation assigns an estimated probability of the same order as that estimated for species seen once. Nyn represents a proper name having a second-person pronoun as head. This is possible – one of us drives to work past a business named *Slender You* – but it seems much more unusual; it is assigned an estimated probability an order of magnitude lower. Njp! represents a noun phrase headed by an adjective, marked as plural, and functioning as an exclamation. Conceivably, someone contemplating, say, the aftermath of a battle might utter the phrase *These dead!*, which would exemplify this species – but the example is patently contrived, and it is assigned an estimated probability much lower still. Thus the probability differentials in these cases do seem to correlate at least very crudely with our intuitive judgements of relative likelihood – certainly better than would be achieved by sharing P_0 equally among the unseen species, which would yield an estimated probability of 1.9e–07 in each case. (As in the bigram case, there are undoubtedly some individual unseen species in this example which will be much rarer or commoner than predicted because of interactions between values of separate parameters.)

This approach to estimating the probabilities of unseen species depends on the nature of particular applications. For most language and speech applications, though, it should be possible to find some way of breaking unseen "species" down into complexes of components or features whose probabilities can be estimated individually, in order to apply this method. For the prosody example, for instance, a particular sequence of sound-classes could be broken down into successive transitions from the eight-member set VC, VR, CC, CR, RC, RR, CV, RV, each of which could be assigned a probability from the available data. Whenever such a technique is possible, it is recommended.

10. CONCLUSION

We have presented a Good-Turing method for estimating the probabilities of seen and unseen types in linguistic applications. This Simple Good-Turing estimator uses the simplest possible smooth of the frequencies of frequencies, namely a straight line, together with a rule for switching between estimates based on this smooth and estimates based on raw frequencies of frequencies, which are more accurate at low frequencies. The SGT method is more complex than additive techniques, but simpler than two-way cross-validation. On a set of Monte Carlo examples SGT proved to be far more accurate than additive techniques; it was more accurate than 2CV for low frequencies, and about equally accurate for higher frequencies.

The main assumption made by Good-Turing methods is that items of interest have binomial distributions. The accuracy tests reported in §7 relate to artificial data for which the items are binomially distributed; how far the usefulness of SGT may be vitiated by breakdowns of the binomial assumption in natural-language data is an unexplored issue.

The complexities of smoothing may have hindered the adoption of Good-Turing methods by computational linguists. We hope that SGT is sufficiently simple and accurate to remedy this.

NOTES

1. Many writers reserve the term *frequency* for counts of specific items, and if we conformed to this usage we could not call p_i a "population frequency": such a quantity would be called a "population probability". In order to make the present paper accessible to its intended audience, we have preferred not to observe this rule, since it leads to conflicts with ordinary

English usage: people speak of the frequency, not the probability, of (say) redheads in the British population, although that population is open-ended, with new members constantly being created. Furthermore I.J. Good, the writer whose ideas we shall expound, himself used the phrase "population frequency". But it is essential that readers keep distinct in their mind the two senses in which "frequency" is used in this article.

2. The *likelihood* of x given y is the probability of y given x, considered as a function of x (Fisher, 1922: 324-327; Box & Tiao, 1973: 10). The maximum likelihood estimator selects that population frequency which, if it obtained, would maximize the probability of the observed sample frequency. That is not the same as selecting the population frequency which is most probable, given the observed sample frequency (which is what we want to do).
When reading Church & Gale (1991), in which the item cited above is an appendix, one should be aware of two notational differences between that article and Good (1953), to which the notation of the present article conforms. Church & Gale use N_r rather than n_r to represent the frequency of frequency r; and they use V (for "vocabulary") rather than s for the number of species in a population.
Mosteller & Wallace (1964) concluded that for natural-language data the "negative binomial" distribution tended to fit the facts better than the binomial distribution; however, the difference was not great enough to affect the conclusions of their own research, and the difficulties of using the negative binomial distribution have in practice kept it from being further studied or used in the subsequent development of computational linguistics, at least in the English-speaking world.
When r is the highest sample frequency, Z_r is computed by setting r'' to a hypothetical higher frequency which exceeds r by the same amount as r exceeds r'.
To avoid confusion, we should point out that Zipf made two claims which are *prima facie* independent (Zipf, 1935: 40-48). The generalization commonly known as Zipf's Law (though Zipf himself yielded priority to J.B. Estoup) is that, if vocabulary items or members of analogous sets are ranked by frequency, then numerical rank times frequency is roughly constant across the items. This law (later corrected by Benoît Mandelbrot, cf. Apostel et al., 1957) does not relate directly to the discussion above, which is not concerned with rank order of species. But Zipf also held that frequency and frequency-of-frequency are related in a log-linear manner. In fact Zipf claimed (see e.g. Zipf, 1949: 32, 547 n. 10 to ch. 2) that the latter generalization follows from the former; however, we do not rely on (and do not accept) this argument, pointing out merely that our empirical finding of log-linear relationships in diverse language and speech data-sets agrees with a long-established general observation.
7. Statistically sophisticated readers might expect data points to be differentially weighted in computing a line of best fit. We believe that equal weighting is a good choice in this case; however, a discussion would take us too far from our present theme.
8. The implementations of the SGT technique reported later in this paper used the coefficient 1.65 rather than 1.96, corresponding to a .1 significance criterion. This means that there are likely to be a handful of cases over the range of examples discussed where a p_r estimate for some value of r was based on a raw proxy where, using the more usual .05 significance criterion, the technique would have selected an estimate based on a smoothed proxy for that particular value of r.
9. The approximations made to reach this are that n_r and n_{r+1} are independent, and that $Var\ (n_r) \cong n_r$. For once the independence assumption is reasonable, as may be gathered from how noisy n_r is. The variance approximation is good for binomial sampling of species with low probability, so it is consistent with Good-Turing methodology.
10. Standard smoothing methods applied over the entire range of r will typically oversmooth n_r for small r (where the unsmoothed data estimate the probabilities well) and undersmooth n_r for large r (where strong smoothing is needed); they may also leave local minima and maxima, or at least level stretches, in the series of smoothed n_r values. All of these features are unacceptable in the present context, and are avoided by the SGT technique. They are equally avoided by the smoothing method used in Church & Gale (1991), but this was sufficiently complex that neither author has wished to use it again.
11. Chinese script represents morphemes as units, and lacks devices comparable to word-spacing and hyphenation that would show how morphemes group together into words.
12. Testing the statistical significance of such a difference is an issue beyond the scope of this paper.
13. Source code implementing this algorithm is available by anonymous ftp from ftp.cogs.susx.ac.uk, in file /pub/users/geoffs/SGT.c.
14. Regression analysis yields some line for any set of data points, even points that do not group round a linear trend. The SGT technique would be inappropriate in a case where r and Z were not in a log-linear relationship. As suggested in §4, we doubt that such cases will be encountered in the linguistic domain; if users of the technique wish to check the linearity of the pairs of values, this can be done by eye from a plot, or references such as Weisberg (1985) give tests for linearity and for other ways in which linear regression can fail.
15. Katz (1987) used an estimator which approximates to the Good-Turing technique but incorporates no smoothing: r^* is estimated as $(r+1)\frac{n_{r+1}}{n_r}$ for values of r below some number such as 6 chosen independently of the data, and simply as r for higher values of r, with renormalization to make the resulting probabilities sum to one. Although having less principled justification than true Good-Turing meth-

ods, this estimator is very simple and may well be satisfactory for many applications; we have not assessed its performance on our data. (We have also not been able to examine further new techniques recently introduced by Chitashvili & Baayen (1993).)

16. Following Fisher (*loc. cit.*), Box & Tiao (1973: 34-36) give a non-informative prior for the probability, π, of a binomially distributed variable. Their equation 1.3.26 gives the posterior distribution for the probability π after observing y successes out of n trials. The expected value of this probability can be found by integrating π times the equation given from zero to one, which yields $\frac{y+\frac{1}{2}}{n+1}$. This is equivalent to adding one-half to each of the number of successes and failures. Add-Half is sometimes called the "expected likelihood estimate", parallel to the "maximum likelihood estimate" defined above.
17. I.J. Good (who defined one of the empirical additive estimators surveyed by Fienberg & Holland – cf. Good, 1965: 23–29) has suggested to the present authors that additive techniques may be appropriate for cases where the number of species in the population is small, say fewer than fifty (for instance when estimating frequencies of individual letters or phonemes), and yet some species are nevertheless unrepresented, or represented only once, in the sample. This would presumably be a quite unusual situation in practice.
18. The SUSANNE Corpus is available by anonymous ftp from ota.ox.ac.uk, in directory /pub/ota/public/susanne.

REFERENCES

Apostel, L., Mandelbrot, B., & Morf, A. (1957). *Logique, langage, et théorie de l'information*. Paris: Presses Universitaires de France.

Bachenko, Joan, & Gale, W.A. (1993). A corpus-based model of interstress timing and structure. *Journal of the Acoustic Society of America 94,* 1797.

Box, G.E.P., & Tiao, G.C. (1973). *Bayesian Inference in Statistical Analysis*. London: Addison-Wesley.

Chao, Y.R. (1968). *A Grammar of Spoken Chinese*. Berkeley and Los Angeles: University of California Press.

Chitashvili, R.J., & Baayen, R.H. (1993). Word frequency distributions. In: Hřebíček, L., Altmann, G. (eds.), *Quantitative Text Analysis* (*Quantitative Linguistics*, vol. 52). Trier: Wissenschaftlicher Verlag, 54-135.

Church, K.W. (1989). A stochastic parts program and noun phrase parser for unrestricted text. *IEEE 1989 International Conference on Acoustics, Speech, and Signal Processing*, 23-26 May, Glasgow.

Church, K.W., & Gale, W.A. (1991). A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language 5,* 19-54.

Church, K.W., Gale, W.A., & Kruskal, J.B. (1991). The Good-Turing theorem. In: Church, K.W. & Gale, W.A., A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language 5,* 19-54: Appendix A.

Efron, B., & Thisted, R. (1976). Estimating the number of unseen species: How many words did Shakespeare know? *Biometrika 63,* 435-447.

Fienberg, S.E., & Holland, P.W. (1972). On the choice of flattening constants for estimating multinomial probabilities. *Journal of Multivariate Analysis 2,* 127-134.

Fisher, R.A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, A, 222,* 309-368; reprinted in J.H. Bennett (ed.), *Collected Papers of R.A. Fisher, vol. I, 1912-24*, University of Adelaide Press.

Fisher, R.A., Corbet, A.S., & Williams, C.B. (1943). The relation between the number of species and the number of individuals in a random sample of an animal population. *Journal of Animal Ecology 12,* 42-58.

Gale, W.A. & Church, K.W. (1994). What is wrong with adding one? In: Oostdijk, N., & de Haan, P. (eds.), *Corpus-Based Research into Language*. Amsterdam: Rodopi, 189-198.

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

Good, I.J. (1965). *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. Cambridge, Mass.: M.I.T. Press.

Good, I.J., & Toulmin, G.H. (1956). The number of new species, and the increase in population coverage, when a sample is increased. *Biometrika 43,* 45-63.

Goodman, L.A. (1949). On the estimation of the number of classes in a population. *Annals of Mathematical Statistics 20,* 572-579.

Hinsley, F.H., & Stripp, A. (eds.) (1993). *Codebreakers: The Inside Story of Bletchley Park.* Oxford: Oxford University Press.

Hodges, A. (1983). *Alan Turing: The Enigma of Intelligence*. London: Burnett Books.

Jeffreys, H. (1948). *Theory of Probability*, 2nd ed. Oxford: Clarendon Press.

Jelinek, F., & Mercer, R. (1985). Probability distribution estimation from sparse data. *IBM Technical Disclosure Bulletin 28,* 2591-2594.

Johnson, W.E. (1932). Probability: the deductive and inductive problems. *Mind* n.s. *41,* 409-423.

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 ASSP-35,* 400-401.

Lidstone, G.J. (1920). Note on the general case of the Bayes-Laplace formula for inductive or *a posteriori* probabilities. *Transactions of the Faculty of Actuaries 8,* 182-192.

McNeil, D. (1973). Estimating an author's vocabulary. *Journal of the American Statistical Association 68,* 92-96.

Marshall, I. (1987). Tag selection using probabilistic

methods. In: Garside, R.G., Leech, G.N., & Sampson, G.R. (eds.), *The Computational Analysis of English*. Harlow, Essex: Longman, 42-56.

Mosteller, F., & Wallace, D.L. (1964). *Inference and Disputed Authorship:* The Federalist. London: Addison-Wesley.

Nádas, A. (1985). On Turing's formula for word probabilities. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-33,* 1414-1416.

Perks, W. (1947). Some observations on inverse probability including a new indifference rule. *Journal of the Institute of Actuaries 73,* 285-312.

Press, W.H., Flannery, B.P., Teukolsky, S.A., & Vetterling, W.T. (1988). *Numerical Recipes in C.* London: Cambridge University Press.

Sampson, G.R. (1995). *English for the Computer: The SUSANNE Corpus and Parsing Scheme.* Oxford: Clarendon Press.

Sproat, R., Shih, C., Gale, W.A., & Chang, N. (1994). A stochastic finite-state word-segmentation algorithm for Chinese. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics,* 66-73.

Weisberg, S. (1985). *Applied Linear Regression,* 2nd ed. London: Wiley.

Zipf, G.K. (1935). *The Psycho-Biology of Language: An Introduction to Dynamic Philology.* London: Houghton Mifflin; reprinted by M.I.T. Press (Cambridge, Mass.), 1965.

Zipf, G.K. (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology.* London: Addison-Wesley; reprinted by Hafner, London, 1965.

APPENDIX

This appendix contains the full set of (r, n_r) pairs for the prosody example of §2.

r	n_r
1	120
2	40
3	24
4	13
5	15
6	5
7	11
8	2
9	2
10	1
12	3
14	2
15	1
16	1
17	3
19	1
20	3
21	2
23	3
24	3
25	3
26	2
27	2
28	1
31	2
32	2
33	1
34	2
36	2
41	3
43	1
45	3
46	1
47	1
50	1
71	1
84	1
101	1
105	1
121	1
124	1
146	1
162	1
193	1
199	1
224	1
226	1
254	1
257	1
339	1
421	1
456	1
481	1
483	1
1140	1
1256	1
1322	1
1530	1
2131	1
2395	1
6925	1
7846	1