Language Learning Through Similarity-Based Generalization

Daniel G. Yarlett and Michael J.A. Ramscar
Psychology Department
Stanford University

Abstract

The idea that language can be learned and processed based on probabilistic information has been the subject of much criticism. Principal among these criticisms is the issue of data-sparsity: it is claimed that the amount of data required to reliably estimate the parameters of any useful probabilistic model far outstrip the amount of language exposure any person could reasonably receive, not just in childhood, but in an entire lifetime. The most acute form of this objection involves unseen events: in any sample of language to which a learner is exposed, many legitimate linguistic constructions will not occur, because language is such a complex, productive system. Simple probabilistic models based on the principle of maximum likelihood will assign these events a probability of 0, incorrectly implying that they are (probabilistically speaking) impossible. In this paper we review the standard arguments regarding data-sparsity and the various methods of probability smoothing that have been proposed in the field of natural language processing in order to address them. We then propose a similarity-based model for estimating bigram probabilities in language, based on psychological principles of similarity-based generalization. In a series of 3 computational simulations we show that this method is capable of learning bigram probabilities more efficiently than other methods, and thus results in considerably improved language modeling performance. We argue that the method we propose, as well as being extremely competitive in engineering terms, also provides a powerful way of ameliorating the problems of data-sparsity, and goes some way to showing how the influential criticisms that have been levied at probabilistic models of human language learning might be solved in a manner consistent with basic psychological principles. We argue that these results, in conjunction with the ubiquity of similarity-based mechanisms in cognition, support the hypothesis that human language learners could in fact exploit similarity-based information when acquiring linguistic structure, and that there is no in principle reason to believe that language is not learned and processed through the exploitation of probabilistic information.

One of the most impressive and difficult to explain abilities that language users possess is their capacity for understanding and producing a seemingly limitless number of utterances, many of which they will never have encountered before. How are people able to do this? Proposals regarding the representations that underpin human language learning in order to allow for this fall into two broad categories. In one, it is proposed that language is based upon a complex system of recursive rules which govern the basic syntax of language, and from which the multitude of surface-forms can be derived or constructed (see Chomsky, 1957, 1965, for some of the earliest and most influential statements of this idea). In the other, language is treated as being based upon probabilistic (or quantitative) propensities of varying strength, that are learned through conditioning on elements of the environment, and in particular on previously observed samples of language (see Skinner, 1957, for one early treatment of this kind of idea, and Chater and Manning, 2006, for a more recent formulation of this kind of probabilistic approach).

The latter proposal, that natural language can be understood as a fundamentally probabilistic or quantitative process, and that linguistic production and comprehension can be cast in terms of learned expectancies on the part of language users has been subjected to many attacks, perhaps the most notable being mounted by Miller and Chomsky (1963; but see also Pinker and Prince, 1988; Pinker, 1994; Gold, 1967). Indeed, the impact of these objections has been so marked that a great deal of recent research into language proceeds on the assumption that language is fundamentally unlearnable, and that therefore a considerable number of the principles that govern it must somehow be innately specified (Chomsky, 1965, 1986; Pinker, 1984, 1994).

In spite of the received reservations regarding probabilistic accounts of language, in recent years there has been a renewal of interest in this approach that is attributable to a variety of causes. These include the manifest difficulty of specifying complete generative grammars for natural language; an increased awareness of the lexical-specificity of many patterns in language that sit less naturally in traditional rule-based syntactic theories (Culicover, 1999; Tomasello, 2000; Goldberg, 2003); an increased appreciation of the probabilistic learning mechanisms available even to young children (Saffran, Aslin and Newport, 1996; Saffran, Newport and Aslin, 1996; Aslin, Saffran and Newport, 1998); and also the increasing application of large-scale computational methods to linguistic analysis (see Manning and Schutze, 1999, and Charniak, 1993, for reviews of such methods).

Although the sophistication of probabilistic models of language has increased considerably over the last few decades, when it comes to considering models of this kind as serious models of human language acquisition, they still seem to be fundamentally plagued by the problem of data-sparsity (especially when syntactic phenomena are under consideration): the objection that in order to train a probabilistic model of language sufficiently complex to capture more than the

---

The authors would like to thank Ewart Thomas especially for his input and support throughout the course of this research. Many thanks are also due to Gordon Bower, Nicolas Davidenko, Ulrike Hahn, David Ho, Adam November, Jon Winawer and Nathan Withoft for comments and suggestions they have made at various points. Daniel Yarlett was supported by a Stanford Graduate Fellowship. This material is based upon work supported by the National Science Foundation under Grant Nos. 0547775 and 0624345 to Michael Ramscar.

most trivial of patterns, vastly more data would be required than can be collected for most practical purposes (and even if the model can be trained on a computer, the amount of data required to do so will nevertheless outstrip that available to human learners by many orders of magnitude, thus undermining the models claim to be a model of human language learning). Indeed, for many theorists, the basic problem of data-sparsity has been accepted as a proof of the impossibility of a probabilistic account of language and its learnability (Gold, 1967; Pinker, 1994).

For example, the class of so-called poverty of the stimulus arguments, perhaps the most convincing and frequently advanced arguments in support of the idea that language is unlearnable from the input, are essentially arguments from data-sparsity. A poverty of the stimulus argument typically asks one to consider some syntactic phenomenon, S, and some children of age X. It is argued that the chance of a child of age X encountering an instance of S in their linguistic input is vanishingly small – essentially there is a problem of data-sparsity for children of age X with respect to S – but that nevertheless children of age X reliably exhibit sensitivity to S either in comprehension or production. The conclusion drawn is that S cannot have been learned by the children in question through induction on the input (because it was never encountered), and that therefore knowledge of S must somehow be innate in the children.

But there are (at least) two grounds for objecting to this form of data-sparsity argument. (1) Is it really true that children of age X never encounter S, whatever it may be? (2) Even if S is never encountered by children of age X, is it not still possible that there are some other constructions, S', that the child has experienced and from which children can generalize in order to show sensitivity to S? Objection (1) revolves around the status of the claims about data-sparsity and, in some of the most common forms of poverty of the stimulus arguments these claims appear, upon closer inspection, to be rather dubious (see Pullum and Scholz, 2002, for some exemplary scholarship concerning this issue). Objection (2) revolves around the mechanisms of generalization available to children in language learning, an aspect of language learning that is strangely often overlooked. Commonly, poverty of the stimulus arguments make the strong assumption that children are incapable of generalization, and that therefore they can only learn about a construction from examples of it and it alone. However, the claim that children are entirely incapable of generalization is a questionable assumption that is at odds with much research (see, for example, Ramscar and Yarlett, 2007; Maslen, Theakston, Lieven, Tomasello, 2004; Pullum and Scholz, 2002; Seidenberg and MacDonald, 1999; Rumelhart and McClelland, 1986; Berko, 1958). Moreover, even if children's ability to generalize turns out to be relatively impoverished, it is still possible that the definition of evidence deemed relevant to S is excessively limited, and excludes much relevant information that is available in the input and can be used by even relatively naive learners.

Based on this analysis, it seems to us that much headway can potentially be made against data-sparsity arguments by considering more closely the types of generalization that children may be able to engage in. In what follows, therefore, we describe the structure of traditional probabilistic language models and how they fall foul of the problem of data-sparsity. We then present a review of previous methods of 'smoothing' – engineering methods that have been proposed in the field of natural language processing in order to combat data-sparsity – explaining the rationale for these methods and their strengths and weaknesses. On the basis of this analysis, we

then present a similarity-based model for learning probabilities in language, as well as a novel methodology for assessing how well language models learn the transitional probabilities of events in language. We then present a series of 3 simulations which compare the performance of our similarity-based method against the other smoothing methods that have been proposed, and show that our model is consistently capable of outperforming them. We conclude by discussing the merits of our similarity-based framework, including ways in which it can be extended to account for higher-order language modeling, and the degree to which the processes on which it is based can be considered psychologically plausible. We conclude by arguing that a similarity-based mechanism of the type we describe could reasonably be exploited by human language learners, and consider the consequences of this for our understanding of human language acquisition.

Language Models and the Problem of Data-Sparsity

A language model provides a way of assigning a probability to a sequence of words which, for convenience, we will refer to as 'sentences' without intending to imply that these sequences need be grammatical in any conventional sense (they could, for example, be elliptical, fragmentary, colloquial, or simply malformed, from the point of view of a prescriptive grammar). Developing a language model can be useful for a number of reasons. From a psychological perspective, such a model could serve as the basis for explanations of both comprehension behavior (for example, garden-path effects could potentially be explained because the incorrect interpretation that people often jump to is highly probable according to the model, whereas the correct interpretation is less probable). A probabilistic model could also potentially offer an explanation of production behavior, or grammatical knowledge more generally: the idea would be to examine whether the probabilities assigned by the model can be used discriminate between 'grammatical' and 'ungrammatical' utterances – if it can, the model can be said to have acquired some degree of grammatical (or syntactic) sensitivity. Probabilistic language models are also used in engineering contexts, both to disambigate potential messages in speech-recognition (if an acoustic signal is ambiguous, then a probabilistic model can be used to mandate selection of the most probable parse; Jelinek, 1998), and also in the process of machine translation. But what exactly does a language model consist of?

The aim of a probabilistic language model is to accurately estimate the probability of arbitrary sequences of words in a language. That is, for a sentence S, composed of words $w_1 w_2 w_3 ... w_N$, we wish to estimate

$$\begin{aligned} P(S) &= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)\cdots P(w_N|w_1 \cdots w_{N-1}) \\ &= \prod_{i=1}^{N} P(w_i|w_1 \cdots w_{i-1}) \end{aligned} \tag{1}$$

From this description it may seem that language modeling is a straightforward process: all one needs to do is estimate the relevant probabilities, and then apply them to samples of language through multiplication according to the above formula. However, the decomposition of probabilities expressed above, although mathematically valid, is of little practical use. According to

Equation 1 the probability of the ith word in a sequence is conditional upon the $i-1$ previous words. This means that for even very short sentences, the probability of a word rapidly becomes conditioned on a considerable number of preceding words. To see why this is a problem, we have to examine how it is that probabilities are standardly estimated from empirical counts. The maximum likelihood principle is the most common way of estimating probabilities from empirical data. In the context of language modeling, this principle is applied as follows:

$$P_{ML}(w_i|w_{i-n+1}\cdots w_{i-1}) \equiv \frac{f(w_{i-n+1}\cdots w_i)}{f(w_{i-n+1}\cdots w_{i-1})} \tag{2}$$

where $f(\cdot)$ represents the frequency of a sequence in the sample of language at hand. From this definition it can clearly be seen that if a sequence has a frequency of 0 then the resulting probability estimate will be 0. And from the definition of a language model given in Equation 1, it can be seen that it will only take a single probability estimate of this type to mean that the probability estimate for the whole sequence is 0. This causes a serious problem for language models based on probabilities estimated in this fashion: because the length of the conditioning history of a language model grows with the length of the sequence in question, the presence of a problematic probability is essentially guaranteed due to the productivity of language. This is the basic problem of data-sparsity.

The standard response to this problem is to instead assume that the probability of a word depends only on the previous n words, rather than the entire preceding history. This assumption is known as a Markov assumption: if one assumes that the probability of a word depends only on the previous n words, then one is said to be assuming a Markov language model of order $n+1$ (these models are also referred to as *ngram* models). Ngram models assert an approximation of the following form:

$$\begin{aligned} P(S) &= \prod_{i=1}^{N} P(w_i|w_1\cdots w_{i-1}) \\ &\approx \prod_{i=1}^{N} P(w_i|w_{i-n+1}\cdots w_{i-1}) \equiv P_n(S) \end{aligned} \tag{3}$$

where n indexes the order of the model. Although these models involve assumptions which are not strictly true (it is easy to generate examples in which the probability of a word depends on more than the immediately preceding n words, for virtually any value of n, as we will see shortly), it does allow the statistical estimation process to get off the ground. However, although the use of Markov models helps with data-sparsity, it does not solve the problem entirely. There are two main reasons that the problem of data-sparsity remains so severe: firstly, Zipf's Law (Zipf, 1935, 1949) implies that the vast majority of word types in a language occur very infrequently, and secondly that the number of probabilities to be estimated increases exponentially with the Markov order of the model. We now discuss these issues.

Zipf's Law is an empirically robust regularity for language which states that the probability of finding a word in a sample is approximately equal to the reciprocal of its rank frequency. More formally, Zipf's Law states that

$$p_r \approx \frac{r^{-s}}{\sum_i i^{-s}} \tag{4}$$

where r is the rank frequency of the word in question, and $s > 0$ is a free parameter that is typically close to 1. Zipf's Law thus states that the rank probability distribution of words follows a power-law. Figure 1 shows the result of plotting the log frequency of words in order of increasing rank frequency as sampled from the British National Corpus, a diverse sample of English language consisting of around 100 million words (Burnard, 2000; Burnage and Dunlop, 1992). As can be seen, in log-log space the relationship is approximately linear, which confirms the basic validity of a power-law relationship between these quantities.

A consequence of Zipf's Law is that probabilities – and equivalently frequencies – in language have a very long-tailed distribution. This means that a relatively few number of word types account for the majority of word tokens and that therefore the vast majority of word types occur with low probabilities. Thus, when one samples language one can expect to get a lot of evidence about a few types of words, but at the cost of being able to acquire relatively little evidence about the majority of word types. This compounds the general problem of data-sparsity: not only will many sequences in language fail to occur in even sizable samples of language, but the distribution of their frequencies is heavily skewed, so that most of the sequences that did occur will have very low frequencies, making probability estimates about these events unreliable.

The second reason that estimating the parameters of a Markov model for language is difficult is because the number of probabilities to be estimated grows exponentially with the length of the conditioning history. Ideally, one wants the length of the conditioning event to be as long as possible, in order to accurately model patterns in language: if aspects of a history are relevant to determining the probability of the next word, then this can be represented in the transition probability; if the entire history is not relevant in determining the probability of the next word, then this can also be reflected in the probability estimate (in this case, the probability estimate would be equal to the estimate based on a shorter sub-history). But as the length of a history grows, so does the number of probabilities that need to be estimated, and at an alarming rate.

To illustrate this basic point, let us consider a simple bigram model which we wish to define over a modestly-sized vocabulary consisting of 20,000 words. This means that there are $20,000^2$ or 400 million possible bigrams of concern in the language or, in other words, 400 million possible probabilities that we are attempting to estimate.[1] Therefore, for each possible bigram to have a chance of occurring once in a corpus it would have to be of the order of 400 million words long.[2]

[1] We wish to estimate $P(W_i = y|W_{i-1} = x)$ for $x \in V$, $y \in V$, where V is the set of all words in our vocabulary. In the case of our example, $|V| = 20,000$ and $|V \times V| =$ 400,000,000.

[2] This is clearly only an approximation, which is subject to error on two counts. Zipf's Law suggests that for practical purposes the sample size required to ensure that all legitimate bigrams have a reasonable chance of occurring would be longer than if all words had approximately equal probabilities, because many of the words in question will have very low probabilities of occurring, and hence the expected time we would have to wait for their occurrence, being the reciprocal

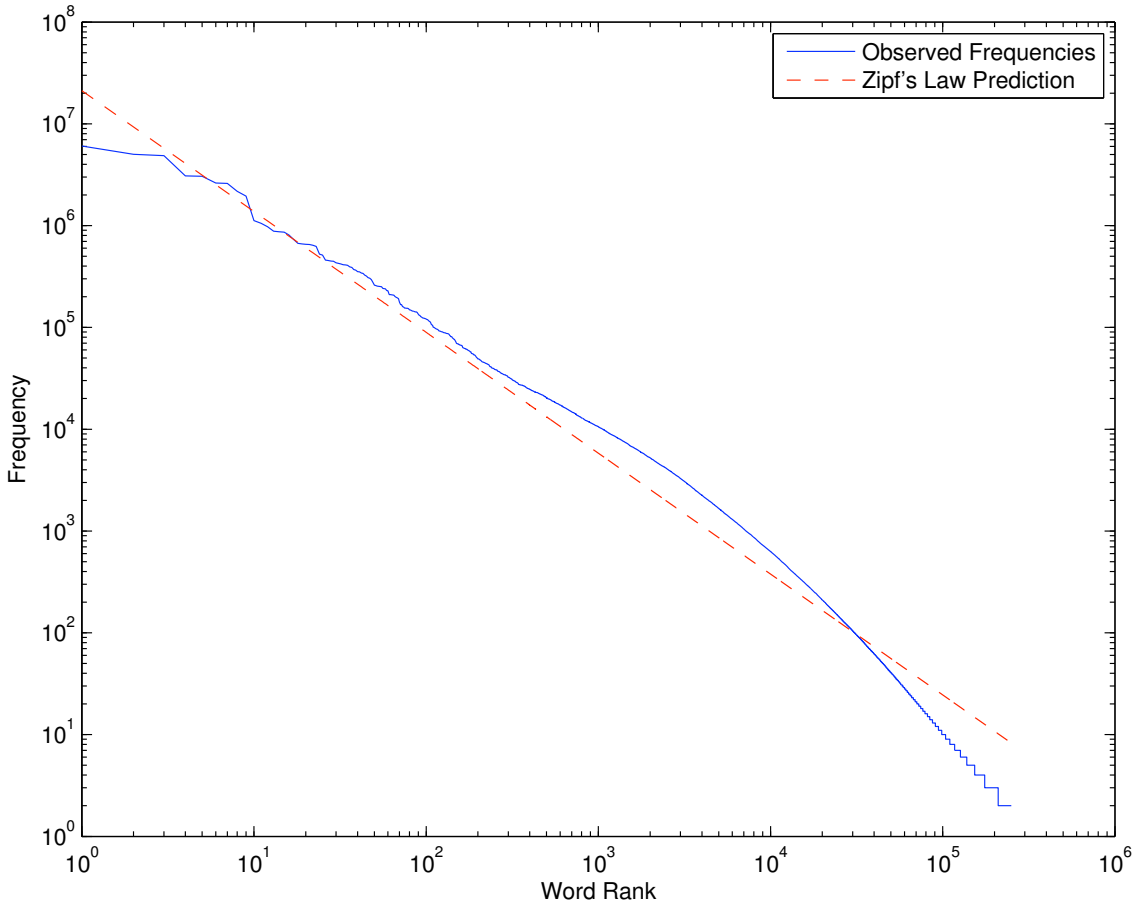

Figure 1. The logarithm of word frequencies plotted against the logarithm of their rank frequency, as extracted from tokens in the British National Corpus. Note the approximately linear relationship between these quantities, which confirms the power-law relationship postulated by Zipf's Law.

And even then, observing each legitimate bigram only once will not be enough to allow for a reliable estimate of the corresponding probability to be reached. One might reasonably expect, then, that a sample of language would need to be hundreds of millions, or even billions, of words long in order to allow all bigram probabilities to be accurately estimated using maximum likelihood.

What happens if we wish to increase the order of our language model? In this case, the problem becomes compounded: we move from attempting to estimate 400 million potential probabilities to attempting to estimate $20000^3 = 8 \times 10^{12}$ potential probabilities. Accordingly, our estimate of the corpus size required to effectively estimate this number of probabilities would increase by several orders of magnitude from that required to estimate the bigram model. In support of this point, Essen and Steinbiss (1992) report that in a 75-25% split of the 1 million word LOB corpus, 12% of the bigrams in the test partition did not occur in the training partition (that is, around 1 in 10 bigrams subsequently encountered had not been seen after training on three quarters of a million words of text). And as we would expect, the problem is considerably worse when we are dealing with trigrams. Brown, Della Pietra, deSouza, Lai and Mercer (1992) examined a corpus consisting of 366 million words, and found that even after this large amount of training data, one can still expect 14.7% of the word triples in any new English text to be absent from the training sample.

Miller and Chomsky (1963) were amongst the first authors to consider this problem in detail, and indeed, their analysis is taken by many as providing a demonstration of the impossibility of a probabilistic account of language. It is thus something of a *locus classicus* when it comes to theorizing about mechanisms for language acquisition, and therefore bears close examination. They pointed out that increasing the order of an ngram model, in order to allow it to represent the intricacies of even a moderate proportion of English sentences, results in there being far too many statistical parameters to be reliably estimated: as n increases, the number of potential parameters to be estimated grows as V^n, where V is the number of tokens in the language. This means that staggering amounts of text are required to allow the probabilities to be reliably estimated:

> "Just how large must n and V be in order to give a satisfactory model? Consider a perfectly ordinary sentence: *The people who called and wanted to rent your house when you go away next year are from California.* In this sentence there is a grammatical dependency extending from the second word (the plural subject *people*) to the seventeenth word (the plural verb *are*). In order to reflect this particular dependency, therefore, n must be at least 15 words. We have not attempted to explore how far n can be pushed and still appear to stay within the bounds of common usage, but the limit is surely greater than 15 words; and the vocabulary must have at least 1000 words. Taking these conservative values of n and V, therefore, we have $V^n = 10^{45}$ parameters to cope with, far more than we could estimate even with the fastest digital computers." (p.430)[3]

of the probability of occurrence, will be extremely long. On the other hand, we would not expect all of the 400 million possible bigrams to be 'legitimate' English bigrams (i.e. to have non-zero frequencies in the limit), and hence probability estimates for these bigrams would not be required, suggesting that the estimate be lowered somewhat.

[3]The notation in this quotation has been changed slightly from the original in order to make it consistent with that in

We can contextualize this (conservative) estimate of the number of parameters that a half-way functional language model would require in a couple of ways. The first is to observe that it has been estimated that a child will only have been exposed to around 50 million written and spoken words by the time they reach the age of 12 (or 5×10^7; Landauer and Dumais, 1997, p.222). The second is to note that the average lifetime consists only of around 2.2×10^9 seconds. Either way, the figure of 10^{45} advanced by Miller and Chomsky outstrips these values by an astronomical margin!

These two reasons – Zipf's Law and the exponential growth in the number of probabilities to be estimated as the history length grows – guarantee that an ngram model, when its probabilities are estimated by maximum likelihood, will assign a probability of 0 to many perfectly legitimate ngram transitions, simply because the appropriate transitions were not encountered in the language sample employed. As Miller and Chomsky (1963) put it: "We know that the sequences produced by *n*-limited Markov sources cannot converge on the set of grammatical utterances as *n* increases because there are many grammatical sentences that are never uttered and so could not be represented in any estimation of transitional probabilities." (p.429) Another way of phrasing this difficulty is that ngram models based on maximum likelihood estimation, if regarded as generative mechanisms, will tend to suffer from a problem of radical undergeneration: from the structure of these models it follows that if a specific transition is unobserved that it forms no part of the language, which is clearly not always, or even most of the time, a valid inference. How can this problem be addressed?

One avenue by which data-sparsity can be addressed concerns the observation that maximum likelihood estimation – the familiar method of counting the number of 'successes' and dividing by the sum of 'successes' and 'failures' that we have been discussing so far – treats every probability as entirely independent of every other. This is exactly analogous to the fictional child in typical poverty of the stimulus arguments who is constitutionally incapable of generalizing from his or her past experience in any meaningful way. Perhaps, if people are estimating the probabilities of linguistic events, they are utilizing a more sophisticated method of probability estimation, one which acknowledges that some probability estimates may be relevant to the values of other probability estimates – in other words, a probability estimation procedure that exhibits some potential to *generalize*. Could such a mechanism offer some headway against data-sparsity?

This is the kind of approach that researchers in the field of natural language processing (NLP) have been pursuing for some time now. Many methods have been proposed to allow the probability of events in language to be better estimated, and these methods of 'smoothing' typically work by combining probability estimates derived from maximum likelihood with information from other relevant probabilities (the methods differing in what is counted as relevant information, and how the information is integrated; see Manning and Schütze, 1999, for a survey of the general strategy). In this fashion, what might begin as 400 million independent probabilities can be treated as a smaller number of somewhat dependent probabilities, and because a smaller number of probabilities needs to be estimated, the hope is that the amount of data required to do so can be appreciably reduced. We now review the most common smoothing methods that have been proposed in order to see what

---

the rest of this paper.

can be learned from them.

Existing Smoothing Methods

Smoothing methods are methods which, given a sample of language, are designed to arrive at more accurate estimates of the probabilities of linguistic events than simple maximum likelihood (see Equation 2). Because language is plagued by the problem of data-sparsity, the methods we describe below have been proposed in order to provide a way of converging on the population probability of an event more rapidly – with a smaller sample size – than the unadjusted empirical probabilities reached through maximum likelihood. In what follows, we describe the most common and successful smoothing methods and how they can be applied in the particular case of estimating bigram conditional probabilities.[4]

Notation

Before we describe the various smoothing methods it will be convenient to establish some notation.

- w_i denotes the word in the i^{th} position in a sequence. Occasionally $w_i \cdots w_j$, where $j > i$, will be used to denote the words running from the i^{th} to the j^{th} position (inclusive) in a sequence. Where a subscript is less than or equal to 0, for example w_{i-n} where $n \geq i$, it is assumed to refer to a null element.
- $f(\cdot)$ denotes the frequency of the specified sequence in a given corpus of text (usually the training corpus).
- V is the set of vocabulary words, or the words that we wish our model to be defined over; $|V|$ is thus the size of the vocabulary being used to construct a given language model.
- n_r denotes the number of bigrams with a frequency of r in the corpus. For example, n_0 would represent the number of bigrams which have a frequency of 0 in the sample (i.e. which failed to occur in the sample at all).
- The symbol '$\bullet$' is used to generalize across words. Thus, 'the $\bullet$' refers to the sequences consisting of the word 'the' followed by any other word in V.
- $N_{1+}(w_{i-n+1} \cdots w_{i-1}\bullet) \equiv |\{w_i : f(w_{i-n+1} \cdots w_{i-1}w_i) > 0\}|$. This is thus the number of distinct word types that have been observed to follow the specified history $(w_{i-n+1} \cdots w_{i-1})$ in the corpus.
- $N_{1+}(\bullet\bullet) = \sum_{w_{i-1}} N_{1+}(w_{i-1}\bullet) = |\{(w_{i-1}, w_i) : f(w_{i-1}w_i) > 0\}| = \sum_{w_i} N_{1+}(\bullet w_i)$. This is thus the number of unique bigrams with a frequency greater than 0 in the corpus (numerically equal to $|V|^2 - n_0$).

[4] Although the various methods can be generalized to higher-orders of language model fairly readily, in this paper we will be concerned mainly with the problem of bigram estimation, and so for descriptive simplicity we focus on this case in our exposition.

Uniform Smoothing

The first smoothing method we discuss cannot really be thought of as a genuine attempt to address the problem of data-sparsity, but nevertheless provides a useful baseline measure of performance. This method simply involves treating every possible conditional probability as being of equal probability. Its definition, therefore, is

$$P_{UN}(w_i|w_{i-1}) \equiv \frac{1}{|V|} \tag{5}$$

While this is clearly a suboptimal procedure – for example, the flat distribution it produces is independent of any observed data and is strongly at odds with what would be expected by Zipf's Law – it does provide a lower bound on performance that helps to place the performance of the other methods in a useful context.

Additive Smoothing

One of the simplest methods that has been proposed to combat data-sparsity and more particularly the problem of 0-frequencies, is additive smoothing. Additive smoothing simply involves adding a small positive value to each bigram frequency count in order to guarantee that none of the frequencies will have a value of 0 (this, in turn, guarantees that the probability of a sequence as calculated by Equation 3 will not be degenerate). A common version of this technique involves adding 1 to each frequency count (this method has been advocated by Lidstone, 1920; Johnson, 1932; and Jeffreys, 1948), as follows

$$P_{+1}(w_i|w_{i-1}) \equiv \frac{f(w_{i-1}w_i) + 1}{f(w_{i-1}) + |V|} \tag{6}$$

While this method guarantees that no probability will be assigned a value of 0, it does have the undesirable consequence that the probabilities of all events, including those we would never expect to observe in a corpus, alike have their frequencies increased. This would not be a problem if there are likely to be only a few bigrams with 0 frequencies, but in fact it is typical for there to be many such bigrams. A consequence of this is that a large amount of the available probability mass is redistributed to unseen events, meaning that the estimated probabilities of observed events becomes systematically underestimated. Gale and Church (1990; 1994) argued that, in general, the amount of probability mass redistributed to bigrams with 0 frequencies by the add-one method is unacceptably large. This means that, for practical applications, the add-one method tends to perform very poorly.

An obvious refinement of the add-one method is to increment the count of bigrams with a frequency of 0, but by a smaller amount. This leads to the simple generalization of add-one smoothing, which we refer to as add-delta smoothing:

$$P_{+\delta}(w_i|w_{i-1}) \equiv \frac{f(w_{i-1}w_i) + \delta}{f(w_{i-1}) + \delta|V|} \tag{7}$$

where $0 < \delta < 1$. This method still has the disadvantage that all events with a frequency of 0 are treated equally, when we might reasonably expect some of them to be more likely than others, but because the increment term may be smaller than 1, the problem of allotting a disproportional amount of probability mass to unseen events is reduced. However, this comes at the cost of having to estimate an appropriate value for the free parameter δ.

Good-Turing Method

The Good-Turing method (Good, 1953) embraces the idea that the observed frequencies of events, particularly when data is sparse, do not give a reliable guide to their true population probabilities, and seeks to estimate what these real probabilities might be based on the observed data and some assumptions about the process by which that data was generated (it is thus an example of an 'empirical Bayes' method). In particular, the Good-Turing method can be used to address the 'missing species' problem: if we take a small sample from a population, how many types of object (e.g. words) do we fail to observe because of the sparsity of our sample? Or, in other words, what is the real size of the vocabulary of a language, given the number of word types we found in a sample of a given size?

Because the Good-Turing method does not involve mixing higher- and lower-order probability distributions – for example, supplementing the potentially sparse information about a conditional bigram distribution with information from the corresponding unigram distribution, as in the case of the more sophisticated smoothing methods we will review later – it necessarily treats all events with a frequency of 0 equivalently. As a result of this, the Good-Turing method cannot be expected to provide estimates for bigram probabilities as accurately as some of the later methods we will discuss, which take into account the specific identity of the words involved rather than just their frequency. Nevertheless, the Good-Turing method is used as a foundation for many of the other smoothing methods we will later discuss, and in addition arises as the result of an interesting statistical analysis, and hence merits its own exposition.

The Good-Turing method asks what is the *expected* frequency of an event (in our case, the occurrence of a bigram) given the data we have observed? Under some ancillary assumptions the answer to this question can be found exactly. The result of this is that one should treat every event that occurred r times in a given sample as though it actually occurred r^* times, where

$$r^* \approx (r+1)\frac{E_{N+1}(n_{r+1})}{E_N(n_r)} \tag{8}$$

(recall that n_r denotes the number of events in our sample with a frequency of r). If we define,

$$g(x) \equiv (f(x)+1)\frac{n_{f(x)+1}}{n_{f(x)}} \tag{9}$$

then we can write the estimate of a conditional bigram probability under Good-Turing as

$$P_{GT}(w_i|w_{i-1}) \equiv \frac{g(w_{i-1}w_i)}{\sum_x g(w_{i-1}w_x)} \tag{10}$$

A mathematical derivation of the Good-Turing method is presented in Appendix A.

The Good-Turing formula as written in Equation 8 cannot, as it stands, be applied to a language sample: it defines r^* in terms of the *expected* value of frequency of frequency counts (the n_rs), and not the actual counts (which are derived from a language sample). The simplest way to proceed is to assume that the actual values are reliable estimates of the expected values, and simply to plug these quantities into the equation. If n_r is substituted for $E_N[n_r]$ in this fashion, then the resulting frequencies are known as the Turing estimators for r^*. However, data-sparsity tends to make this approach problematic. While there are usually many distinct types with low frequencies, as the frequency in question becomes larger the number of types with this frequency becomes fewer and fewer (for example, there will be many words in a sample with a frequency of 1, but very few with a frequency of 10,237; this situation is a natural consequence of the long-tailed Zipfian distribution that language exhibits). Eventually the variance of these values becomes very high meaning that they are unreliable, or 0 counts are observed which cause the adjusted estimates of Equation 8 to be degenerate. It is therefore a common practice to use a smoothed version of the n_r values, $S(n_r)$, following Good (1953). When this method is adopted, the resulting frequencies are known as Good-Turing estimators. Note, however, that there are many potential ways in which the n_r values can be smoothed, and one must therefore be careful to distinguish between these methods. We briefly survey a couple of the most common Good-Turing smoothing methods below.

Church and Gale (1991) propose that frequencies of frequencies (the empirical n_r counts) equal to 0 can be smoothed through linear interpolation between the nearest count on either side with a non-zero value. For example, if $n_{19} = 4$, $n_{20} = 0$, and $n_{21} = 2$, then we would set n_{20} to $(4 + 2)/2 = 3$. Church and Gale also propose enhancing the Good-Turing method by clustering bigrams together based on their probability assuming that the two words are independently distributed of one another. They thus divide bigrams into distinct classes based on their probability under independence, and run the Good-Turing process on these classes separately (i.e. the frequency of frequency counts are kept separately for each class). They report that this leads to an improvement in smoothing performance over the straightforward Turing estimators.

Gale (1995) outlines an alternative method of Good-Turing smoothing, which results in what is known as the Simple Good-Turing (SGT) method. The Simple Good-Turing method proposes that the raw frequency of frequency counts be used as long as they are significantly different from the values predicted by a linear regression model of $\log n_r$ on $\log r$. The rationale behind this is that the raw n_r counts are likely to be accurate when r is small because these values will be based on a large number of data-points. However, as r becomes larger, fewer and fewer events will be observed, and therefore some kind of smoothing will be required. Empirical and simulation studies show that, as a direct consequence of Zipf's law, these two quantities tend to have an almost perfectly linear relationship in log-log space, and therefore a linear regression model produces accurate estimates of the true frequency of frequency counts in a robust fashion. The Simple Good-Turing method uses the following approximate variance estimate:

$$\text{Var}(r^*_{GT}) = (r+1)^2 \left(\frac{n_{r+1}}{n_r^2}\right)\left(1 + \frac{n_{r+1}}{n_r}\right) \tag{11}$$

The Turing estimates are considered to be "significantly different" from the linear regression estimates when the difference between corresponding values exceeds 1.65 times the standard deviation of the statistic (derived from the variance formula above). The Simple Good-Turing method has the advantage that estimates of Good-Turing corrected frequencies can be derived based on much more specific classes of data, because a parametric form for the frequency of frequency counts has been assumed. In general, the SGT method appears to perform extremely well relative to other forms of the Good-Turing approach, and is also readily implemented. It is therefore the main Good-Turing method that we concentrate on in our simulations.

Backoff Smoothing

Backoff smoothing is quite similar to Jelinek-Mercer smoothing (described below), in that it assumes that the unigram probability of an item will be somewhat correlated with its conditional bigram probability, and that therefore the unigram probability can be used as a surrogate for the bigram probability when this estimate is likely to be inaccurate. Therefore, if the bigram frequency of an item is greater than 0, it is used directly. However, when the bigram probability of an item is 0, it is replaced by the unigram probability. The backoff replacement values can thus be defined as follows:

$$P_{BO1}(w_i|w_{i-1}) \equiv \begin{cases} P_{ML}(w_i|w_{i-1}) & \text{where } f(w_{i-1}w_i) > 0 \\ P_{ML}(w_i) & \text{where } f(w_{i-1}w_i) = 0 \end{cases} \tag{12}$$

In order to derive the actual probability estimates, the P_{BO1} quantities need to be normalized so that they sum to 1:

$$P_{BO}(w_i|w_{i-1}) \equiv \frac{P_{BO1}(w_i|w_{i-1})}{\sum_x P_{BO1}(W_i = x|w_{i-1})} \tag{13}$$

Jelinek-Mercer Smoothing

The Jelinek-Mercer smoothing algorithm (Jelinek and Mercer, 1980) assumes that the conditional probability of a word is not entirely independent of its marginal probability. In other words, in the absence of direct evidence we might expect $P(\text{you}|\text{ask}) \gg P(\text{thou}|\text{ask})$ because $P(\text{you}) \gg P(\text{thou})$. Based on this assumption, the Jelinek-Mercer smoothing method estimates bigram probabilities through a linear combination of the empirically observed probability and the marginal probability of the event in question:

$$P_{JM}(w_i|w_{i-1}) \equiv \lambda P_{ML}(w_i|w_{i-1}) + (1-\lambda)P_{ML}(w_i) \tag{14}$$

where $0 \leq \lambda \leq 1$. We can see that the free parameter, λ, determines the degree to which the probability estimate depends on the relevant bigram and unigram counts. We would expect that as data-sparsity becomes more of a problem – i.e. as $f(w_{i-1}w_i)$ becomes smaller – that λ should also become smaller. We discuss the manner in which the parameters of smoothing methods can be set later. However, for the present it is worth noting that using a separate value of λ for each

individual bigram leads to a prohibitive number of parameter values which need to be estimated. Therefore, previous approaches have tended to bucket bigrams according either to their frequency (Bahl, Jelinek and Mercer, 1983), or else according to the mean non-zero value for the frequency distribution following w_{i-1} (Chen and Goodman, 1996), and to use a separate λ for each category thus formed.

Witten-Bell Smoothing

Witten-Bell smoothing (Witten and Bell, 1991) is derived from theories of information compression and takes the same basic form as Jelinek-Mercer smoothing, except that the parameter λ is no longer free but set to a specific value based on probabilistic considerations (Witten-Bell smoothing is thus a special case of Jelinek-Mercer smoothing). This avoids the need to estimate the value of a parameter in a potentially costly search process, but also implies that Witten-Bell can never perform better than Jelinek-Mercer smoothing when its free parameter is set through optimization.

In line with the basic form of Jelinek-Mercer smoothing, Witten-Bell smoothing expresses its smoothed conditional probability estimates as a linear combination of the conditional probability of w_i and the unconditional (i.e. marginal) probability of w_i:

$$P_{WB}(w_i|w_{i-1}) = \lambda_{WB} P_{ML}(w_i|w_{i-1}) + (1 - \lambda_{WB}) P_{ML}(w_i) \tag{15}$$

However, the λ parameter in this case has a fixed, rather than a free, value. This value is determined as follows, as a function of the history preceding w_i:

$$\lambda_{WB} \equiv 1 - \frac{N_{1+}(w_{i-1}\bullet)}{N_{1+}(w_{i-1}\bullet) + f(w_{i-1})} \tag{16}$$

This expression can be interpreted as giving an approximate value for the probability that the word occurring after w_{i-1} will be a word that has already been observed to occur in that position. The model thus relies on the lower-order unigram model more when it is more likely that the word occurring after the history is going to be one that has *not* already occurred in that context. This is a sensible procedure because as one gains more and more data about the language in question, the probability of being surprised by a novel word in the position after word w_{i-1} should decrease, and the weighting of the full bigram model should become higher.

A more accurate, but also more costly, estimate of the probability of a novel event can be derived in the following way (Good and Toulmin, 1956; Efron and Thisted, 1976):

$$1 - \lambda_{WB} = \frac{n_1}{f(w_{i-1})} - \frac{n_2}{f(w_{i-1})^2} + \frac{n_3}{f(w_{i-1})^3} - \frac{n_4}{f(w_{i-1})^4} + \cdots \tag{17}$$

where the n_r here are counted over the population of bigrams with w_{i-1} as their initial word (i.e. $w_{i-1}w_1$, $w_{i-1}w_2$, and so on). This estimator was found to provide the best performance in a text compression test out of several alternative methods for estimating λ for use in Witten-Bell smoothing (Witten and Bell, 1991). It can also be approximated at a reduced computational cost by truncating the sum at an earlier stage; this is also likely to avoid the effects of data-sparsity in the n_r counts and avoid the need to smooth them.

Katz Smoothing

Katz smoothing (Katz, 1987) proposes that the estimated probabilities of bigrams should be arrived at in a two-stage process. Bigrams with a frequency greater than 0 should have their frequencies reduced by a *discount ratio*, resulting in a certain amount of frequency mass being reserved. This reserved frequency mass is then redistributed to items with a frequency of 0, in proportion to their marginal probability. For a bigram $w_{i-1}w_i$ with frequency $f(w_{i-1}w_i) = r$, Katz smoothing proposes that we calculate its corrected count using the following scheme:

$$r_{KZ}(w_{i-1}w_i) = \begin{cases} d_r r & \text{if } r > 0 \\ \alpha(w_{i-1}) P_{ML}(w_i) & \text{if } r = 0 \end{cases} \tag{18}$$

where d_r represents the degree to which an item with a frequency of r should have its frequency discounted ($0 \leq d_r \leq 1$ is thus a proportion), and $\alpha(w_{i-1})$ denotes a normalizing constant which is defined below.

The discount ratio d_r represents the degree to which items with non-zero frequencies should have their frequencies discounted in order to free-up some amount of frequency mass for redistribution to those items which had a frequency of 0 in the sample. Katz (1987) proposes that when $r > k = 5$ the empirical frequency counts for these items can be taken as reliable and hence should not be discounted at all (in other words, he assumes that $d_r = 1$ for $r > 5$). For frequencies below this threshold, the discount ratio is chosen to be proportional to the discounts predicted by the Good-Turing method (r^*/r), and such that the sum of the frequencies discounted in the bigram distribution equals the total sum of frequencies that would be assigned to bigrams with a frequency of 0, according to the Good-Turing method. These two constraints are sufficient to uniquely determine the appropriate discount ratios, which are defined as follows:

$$d_r \equiv \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \tag{19}$$

Once the non-zero frequencies in the bigram distribution have been adjusted in the manner described, it only remains to adjust the zero frequencies. This, in turn, requires that the function $\alpha(\cdot)$ be defined (see Equation 18).

The value of $\alpha(w_{i-1})$ is set so that the total number of observations in the distribution does not change. From this constraint, Katz (1987) derives the following definition:

$$\alpha(w_{i-1}) \equiv \frac{1 - \sum_{w_i : f(w_{i-1}w_i) > 0} P_{KZ}(w_i | w_{i-1})}{1 - \sum_{w_i : f(w_{i-1}w_i) > 0} P_{ML}(w_i)} \tag{20}$$

This finally allows all components of an empirical bigram distribution to be adjusted. These adjusted frequencies can then be used in the standard manner in order to provide an estimate of a

conditional probability for use in a language model:

$$P_{KZ}(w_i|w_{i-1}) \equiv \frac{r_{KZ}(w_{i-1}w_i)}{f(w_{i-1})} \tag{21}$$

Katz smoothing has been shown to perform competitively relative to other methods of smoothing (Katz, 1987; Chen and Goodman, 1996, 1998).

Absolute Discounting

Absolute Discounting (Ney, Essen and Kneser, 1994) is an alternative method of smoothing which, instead of combining higher- and lower-order frequency or probability distributions in a linear fashion – as in the case of Jelinek-Mercer smoothing, for example – proceeds by subtracting a fixed discount $D \leq 1$ from the frequency of each item with a non-zero frequency. Because the frequency subtracted from the non-zero frequencies is a constant amount, the proportion subtracted from increasing frequencies diminishes, and this scheme achieves something similar to the decreasing discount ratios proposed by Katz (1987). The basic framework of Absolute Discounting can be specified as follows:

$$P_{AD}(w_i|w_{i-1}) \equiv \frac{\max\{f(w_{i-1}w_i) - D, 0\}}{f(w_{i-1})} + (1 - \lambda)P_{ML}(w_i) \tag{22}$$

As can be seen, the adjusted frequency distribution is supplemented through combination with the maximum-likelihood unigram distribution. The Absolute Discounting framework thus has two parameters that need to be set – D and λ – before definite probability estimates can be arrived at.

The frequency discount parameter, D, can either be set according to the following prescription according to Ney, Essen and Kneser (1994), or else it can be treated as a free parameter to be estimated through a search process:

$$D \equiv \frac{n_1}{n_1 + 2n_2} \tag{23}$$

where n_r is the number of bigrams with a frequency of r, as usual. The λ parameter can also be optimized through a search process, or else set to the following value that Ney, Essen and Kneser (1994) propose:

$$1 - \lambda = \frac{D}{f(w_{i-1})} N_{1+}(w_{i-1}\bullet) \tag{24}$$

One way of interpreting this quantity is as a scaled (by D) estimate of the type:token ratio of the distribution following w_{i-1}, which provides an estimate of the probability that a novel token will be encountered on the next sample from the distribution (Chen and Goodman, 1998). Obviously, if the probability of encountering a new word type after w_{i-1} is high then a larger proportion of the unigram distribution should be mixed into the discounted bigram distribution, which is exactly what occurs under Absolute Discounting.

Kneser-Ney Smoothing

Kneser-Ney smoothing is virtually identical to Absolute Discounting, except that whereas Absolute Discounting combines the bigram probability distribution with the unigram distribution in order to 'fill in' potentially sparse values, the Kneser-Ney method relies on an alternative type of distribution to perform this function. The basic Kneser-Ney framework can be stated in the following way, in accordance with its similarity to Absolute Discounting:

$$P_{KN}(w_i|w_{i-1}) \equiv \frac{\max\{f(w_{i-1}w_i) - D, 0\}}{f(w_{i-1})} + (1 - \lambda)P_{KN}(w_i) \tag{25}$$

As can be seen, this method is the same as Absolute Discounting except that the discounted bigram probabilities are supplemented with the probabilities contained in the Kneser-Ney distribution instead of the unigram distribution (hence $P_{ML}(w_i)$ is replaced by $P_{KN}(w_i)$ in the above definition).

Kneser and Ney (1995) motivate the need for an alternative lower-order distribution in the following way. They observe that the lower-order distribution will only have a significant impact on the resulting probability estimates when the bigram distribution contains a large number of types relative to tokens (in other words, when many distinct words have been observed after w_{i-1} relative to the number of times it has been observed; see Equation 24). Kneser and Ney (1995) therefore argue that the lower-order distribution that one relies upon should be optimized for exactly this sort of situation. In this case, the exact token counts may be less robust than the type counts, and so it may be more sensible to base the lower-order distribution on type counts.

An example will make this more concrete. Imagine that 'San Francisco' occurs frequently in a corpus. This means that the unigram probability of 'Francisco' will be relatively high, and thus the probability of it occurring after *any* word will receive a lot of support from a smoothing method such as Jelinek-Mercer or Absolute Discounting. However, this is probably inappropriate as 'Francisco' only ever tends to occur after 'San' – we should not therefore expect it to occur after *any* word w_{i-1} even though it has a large unigram frequency. Kneser and Ney (1995) propose that it would be better to count the number of contexts in which 'Francisco' has been observed to occur or, in other words, to count the number of word types that it has been observed to follow. In an English corpus we might expect this to be 1: 'Francisco' has only ever been observed to occur after 'San' and therefore the Kneser-Ney distribution would, in general, waste relatively little of its probability mass to the occurrence of 'Francisco' when smoothing the distribution based on an arbitrary word w_{i-1}. Working under the constraint that the marginal distribution of the higher-order bigram distribution must match the marginal distribution of the training data allows the following definition of the Kneser-Ney distribution to be arrived at:

$$P_{KN}(w_i) \equiv \frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)} \tag{26}$$

The parameters, λ and D, of Kneser-Ney smoothing can be set exactly as for the Absolute Discounting method.

Modified Kneser-Ney Smoothing

Modified Kneser-Ney smoothing is a refinement of Kneser-Ney smoothing developed by Chen and Goodman (1998). Instead of using a single discount parameter, D, three discount parameters which apply to different empirical frequencies are applied to the raw bigram distribution (the discount parameters apply to frequencies of one, two, and three or more; no discount parameter is required for a frequency of 0, of course). The basic form of the method is therefore the same as Kneser-Ney smoothing, except that the D parameter is a function of the bigram frequency in question, which allows the model to have greater flexibility. Note that the lower-order Kneser-Ney distribution is exactly the same as defined for Kneser-Ney smoothing (see Equation 26). The definition of Modified Kneser-Ney smoothing is thus:

$$P_{MKN}(w_i|w_{i-1}) \equiv \frac{\max\{f(w_{i-1}w_i) - D(f(w_{i-1}w_i)), 0\}}{f(w_{i-1})} + \gamma(w_{i-1})P_{KN}(w_i) \tag{27}$$

The function which sets the actual discounting parameter to be used is defined as follows:

$$D(f) = \begin{cases} 0 & \text{if } f = 0 \\ D_1 & \text{if } f = 1 \\ D_2 & \text{if } f = 2 \\ D_{3+} & \text{if } f \geq 3 \end{cases} \tag{28}$$

Just as a fixed value for D can be found for Kneser-Ney smoothing (Equation 23), fixed values for the three separate discount parameters can also be specified, or else these parameters can be optimized through a parameter search process:

$$\begin{aligned} Y &= \frac{n_1}{n_1 + 2n_2} \\ D_1 &= 1 - 2Y\frac{n_2}{n_1} \\ D_2 &= 2 - 3Y\frac{n_3}{n_2} \\ D_{3+} &= 3 - 4Y\frac{n_4}{n_3} \end{aligned} \tag{29}$$

Finally, the weighting that the lower-order distribution receives is determined by:

$$\gamma(w_{i-1}) = \frac{D_1 N_1(w_{i-1}\bullet) + D_2 N_2(w_{i-1}\bullet) + D_{3+} N_{3+}(w_{i-1}\bullet)}{f(w_{i-1})} \tag{30}$$

Chen and Goodman (1998) compare the Modified Kneser-Ney technique to a wide variety of other smoothing regimes and found that, in general, it offered the best language modeling performance. The Modified Kneser-Ney method therefore represents the current state-of-the-art when it comes to probability estimation in language modeling.

Similarity-Based Smoothing

The smoothing methods considered so far have essentially proceeded by 'filling in' a bigram frequency distribution with a lower-order distribution – typically either the unigram distribution or the Kneser-Ney distribution – in situations in which data-sparsity is likely to have caused the raw bigram frequency counts to be unreliable. A general limitation of these smoothing methods is therefore that the lower-order distribution remains the same irrespective of the conditioning history, w_{i-1}. However, it is not unreasonable to believe that the identity of the preceding word could be used to tailor the lower-order distribution in some fashion. This is what similarity-based approaches to smoothing attempt to do.

Similarity-based approaches to smoothing attempt to find directly observed distributions which are *similar* in some sense to the distribution which needs to be smoothed, and to use these similar distributions to supplement it (they are thus instances of the general class of nearest neighbor algorithms; see Duda, Hart and Stork, 2001). A concrete example will help to show why one might expect this to be a useful strategy. Imagine that we wish to estimate the conditional probability distribution over words following the word 'raccoon', a word which we will assume for the sake of argument we are not very familiar with. Although we will not have encountered the word 'raccoon' in our linguistic experience very often, we might know enough about it to have learned that it is somewhat similar to the following words: 'skunk', 'cat' and 'badger'. Given that these words are similar – in an as yet unspecified sense – to 'raccoon' we might expect that the same types of words can follow 'raccoon' as can follow these other words. In other words, we might infer that if we have seen many instances of 'skunk ran', 'cat ran' and 'badger ran' that $P(\text{ran}|\text{raccoon})$ will be relatively large (such an inference is analogous to similarity- or feature-based inductive inference; Sloman, 1993). Therefore, if we wish to smoothe our estimate of the distribution of words following 'raccoon', it might be appropriate to rely on the raw distributions derived from words *similar* to 'raccoon'. One of the assumptions underlying this approach is that language is at least a somewhat regular system, and hence that these additional distributions could be used to 'pull' the directly observed distribution in an appropriate direction. Of course, such an approach relies entirely on there being a mechanism by which one can determine the distributions which are likely to be most relevant when it comes to filling in the directly observed information – in other words, one must be able to specify an appropriate form of similarity that can be used. If the wrong distributions are used to supplement a raw distribution, this will simply result in noise being added to the directly observed distribution, which is unlikely to improve language modeling performance (certainly not compared to the smoothing techniques described above). We will discuss this issue shortly, but first we wish to briefly consider some of the attractive properties of a similarity-based approach to smoothing, over and above the intuitive motivation for it just given.

The first potential advantage that similarity-based smoothing methods could have, as we have already indicated, is their great flexibility. Similarity-based approaches are not constrained to using only a single distribution to supplement bigram frequency distributions, and the fact that the lower-order distribution can be tailored to each conditioning history means that this general approach has the potential to offer high levels of language modeling performance. This possibility

alone merits further consideration of the basic approach. In addition, another attractive aspect of similarity-based approaches to smoothing is that similarity-based generalization appears to be a ubiquitous form of human inference which has been studied in some detail, both as a topic in itself (see, for example, Hahn and Ramscar, 2001; Tversky, 1977), and in terms of its involvement in other cognitive processes such as categorization (Nosofsky, 1986), induction (Shepard, 1987; Sloman, 1993), and analogical reasoning (Ramscar and Yarlett, 2003; Gentner and Markman, 1997; Holyoak and Koh, 1987). Given the central role that similarity appears to play in human cognition, it is reasonable to hypothesize that a similarity-based mechanism may also be involved in aspects of human language learning.

Previous Similarity-Based Research

The most notable previous application of a similarity-based approach to language modeling is the model proposed by Dagan, Lee and Pereira (1999). Their model uses a back-off framework, which means that only bigrams with an observed frequency of 0 are adjusted through similarity-based information. The probability mass that is assigned through a similarity-based mechanism is the probability mass freed up by discounting the non-zero bigrams frequencies according to the Good-Turing method. The basic structure of their model can thus be written:

$$P_{DLP}(w_i|w_{i-1}) \equiv \begin{cases} P_{GT}(w_i|w_{i-1}) & \text{where } f(w_{i-1}w_i) > 0 \\ \alpha(w_{i-1})P_S(w_i|w_{i-1}) & \text{where } f(w_{i-1}w_i) = 0 \end{cases} \tag{31}$$

where $\alpha(w_{i-1})$ denotes a normalization constant and $P_S(\cdot)$ is a probability estimated through the similarity-based algorithm. When a probability $P_S(w_i|w_{i-1})$ needs to be estimated using similarity-based information (i.e. when $f(w_{i-1}w_i)$ is 0), the model first finds the distributions $P_{ML}(W_i|W_{i-1} = x)$ where $x \in V$ which are most similar to the distribution $P_{ML}(W_i|W_{i-1} = w_{i-1})$. In order to measure the similarity between distributions Dagan, Lee and Pereira (1999) propose using the Jensen-Shannon metric (this metric is related to the more commonly used Kullback-Leibler metric which will be discussed later, and essentially estimates the amount of informational divergence between two probability distributions). The k nearest words which also have associated distributions that are a distance of less then t from the base distribution are included in a set $S(w_{i-1})$ (denoted as such because the distributions are most similar to the distribution over words following w_{i-1}). Once $S(w_{i-1})$ has been defined, it is used to estimate the probabilities of unseen bigrams in the following way:

$$P_S(w_i|w_{i-1}) \equiv \gamma P_{ML}(w_{i-1}) + (1-\gamma) \sum_{w'_{1-1} \in S(w_{i-1})} \frac{W(w_{i-1}, w'_{1-1})}{\text{norm}(w_{i-1})} P(w_i|w'_{1-1}) \tag{32}$$

where γ is a free parameter controlling the aggressiveness of smoothing, $\text{norm}(w_{i-1})$ is a normalization constant to ensure that the resulting distribution sums to 1, and $W(w_i, w_j)$ is a decreasing function of the distance between word i and word j, that depends on the distance metric being used. For example, for the Jensen-Shannon metric, Dagan, Lee and Pereira propose using the following

weighting function:

$$W(w_{i-1}, w'_{i-1}) \equiv 10^{-\beta J(w_{i-1}, w'_{i-1})} \tag{33}$$

where $J(x, y)$ denotes the Jensen-Shannon divergence between two distributions x and y.

Dagan, Lee and Pereira (1999) report the results of applying their similarity-based model to a standard language modeling task, using text extracted from the Wall Street Journal corpus. Their model uses 4 free parameters – k, t, β and γ – which were set through an informal grid search. In general, using the Kullback-Leibler distance measure, they found that the optimal parameters relied on the nearest 60 ($k = 60$) neighbors, and that 0.15 of the unigram distribution was added to 0.85 of the similarity-based information ($\gamma = 0.15$). On this task, perplexity was reduced by about 20% for unseen bigrams on their test set, compared to a straightforward Katz-like model using the unigram probabilities as the back-off distribution. Because unseen bigrams constituted 10.6% of the test sample, this led to an overall reduction of 2.4% in test-set perplexity: overall perplexity was reduced from 237.4 to 231.7 using a training corpus of 40 million words.

Although Dagan, Lee and Pereira's work demonstrates the potential validity of a similarity-based approach, there are a number of limitations to it. Firstly, the back-off framework they use only adjusts directly observed bigrams with a frequency of 0. However, in the extensive simulations conducted by Chen and Goodman (1996; 1998) the best-performing models were interpolation models, that adjusted the frequency of all bigrams, and not just those with a frequency of 0. Indeed, Dagan, Lee and Pereira admit that "It would also be worth investigating the benefit of similarity-based methods to improve estimates for low-frequency seen events." (1999, p.26) There is good reason to expect, therefore, that implementing a similarity-based model based on interpolated smoothing will lead to performance improvements. Secondly, Dagan, Lee and Pereira did not attempt to vary the values of their free parameters depending on the distribution being smoothed; instead, they used a fixed set of parameters for all distributions, choosing the values that led to the best overall performance. This suggests that similarity-based approaches could perform even better than the work of Dagan, Lee and Pereira suggests, if a closer exploration of the parameter values is undertaken. In addition, Dagan, Lee and Pereira's model is based on Good-Turing estimates, which are used to smooth the unigram distribution they use to 'cover' their similarity-based estimates, which are derived from a relatively complex process (see Appendix A) which is not necessarily cognitively plausible. Finally, and perhaps most importantly, Dagan, Lee and Pereira only report the performance of their similarity-based model relative to Katz smoothing. However, in the most extensive empirical comparison of smoothing methods to date, Chen and Goodman (1996, 1998) found that Absolute Discounting, Kneser-Ney and Modified Kneser-Ney smoothing methods all reliably outperformed Katz smoothing. Therefore, in order to understand how well similarity-based techniques perform relative to the best smoothing methods currently proposed, a more extensive comparison is required. The simulations we report in this paper set out to address these issues.

A Framework for Similarity-Based Smoothing

We now present a framework for similarity-based smoothing. Our goal was to make the framework as general as possible, whilst adjusting directly observed bigram frequencies using an interpolation as opposed to a back-off scheme, so that a range of different similarity-based approaches could be explored and compared to the currently existing smoothing methods we have reviewed. We divide the framework into four basic parts, as follows: (1) the distributional similarity metric used to determine the distributions with which to supplement observed bigram frequencies; (2) the decay function used to determine how the similar distributions are weighted when they are summed across; (3) the definition of the neighboring distribution as a function of the nearest-neighbors and the decay function; and (4) the way in which the supplemental similarity-based information is combined with the directly observed bigram frequencies in order to arrive at a set of smoothed probability estimates. This results in a model which can be configured in a number of ways and which uses 4 free parameters, the same as used by Dagan, Lee and Pereira's (1999) model and the Modified Kneser-Ney model, which currently offers the best level of language modeling performance. We now discuss the model's structure in greater detail under the four headings listed above.

Distributional Similarity Metric.

A similarity-based language model requires some way of finding words which are similar to – or the 'nearest neighbors' of – a given target word. But similar in which respects? A reasonable answer is that the similarity metric used should be sensitive to the syntactic category, or part-of-speech class, to which the target word belongs. The syntactic category of a word has important consequences for the words that can follow it: for example, in English determiners tend to be followed by adjectives or nouns, verbs tend to be followed by determiners, prepositions or nouns, and so on. To give a concrete example, imagine that we wish to smooth the conditional distribution over words given that we have just observed 'children'. In this case, our similarity-metric should only assign other plural nouns a significant degree of similarity to 'children', otherwise the high probability of subsequently observing 'are' but not 'is' would be hard to capture. However, syntactic constraints are not the only relevant ones that should be respected. Semantic constraints can also be relevant, because semantics can serve to constrain verb arguments, adjective choices and the like. For example, children are unlikely to drive or shoot, and hence it might be less than ideal if 'cops' was found to be highly similar to 'children' – in this case, the probability of a word like 'shot' in this context might be over-estimated. Therefore, an optimal similarity-metric to be used in language modeling should respect the traditional syntactic categories established by linguists; but, within these categories, it is also desirable for there to be a semantic gradient too.

How can a similarity measure of this type be derived? One idea which has received an increasing amount of attention in recent years is that distributional information – statistical information about the context in which a word is typically observed to occur – can be used to automatically provide information about the syntactic category of a word, despite earlier arguments that this type of information would be unreliable or too complicated to track (Pinker, 1984). A particularly impressive example of this type of research was presented by Redington, Chater and Finch (1998).

They examined the child-directed speech of adults in the CHILDES corpus (MacWhinney, 2000) and from this data derived representations of certain *target* words in a multidimensional space. They did this by examining all the occurrences of the target words in their corpus, and counting the frequency with which a list of 150 *context* words occurred within a range of 2 words before or after the target word. From this data a representation of each target word was derived, which consisted of a frequency distribution indicating the number of times it was observed to co-occur with each of the 150 context words. These frequency distributions can be thought of as locating the target words within a high-dimensional space, and essentially provide a 'signature' for each target word. Redington, Chater and Finch showed that the Spearman's Rank Correlation between these vectors could be used as a similarity measure between the target words, and that when this is done the resulting distances contain a lot of information about the syntactic category of the target words. For example, by using a hierarchical cluster analysis they showed that the vectors tended to cluster in a fashion which agreed with the syntactic categories posited by linguists. Similar claims about the power of distributional methods have also been made by other researchers (for example, see Cartwright and Brent, 1997; Burgess and Lund, 1997; and Mintz, 2003).

As a result of the manifest success of these models, and the similar approach utilized by Dagan, Lee and Pereira (1999), we decided to employ a distributional method in order to provide a way of generating the nearest neighbors to use in our similarity-based smoothing framework. We decided to use the 20,000 most frequent words in our sample as the context words (they provide the basis of our context space), and to examine the distribution of words which occurred immediately before and immediately after a target word. Given a corpus, this method allowed us to specify two 20,000 element frequency vectors for a given target word, one specifying the frequency distribution occurring before it, and the other specifying the frequency distribution after it. Because of Zipf's Law (Zipf, 1935, 1949) these frequency distributions will be heavily skewed, with a few extremely large frequency counts on average, and many much smaller frequencies. In order to prevent these few values overwhelming the smaller frequency counts and solely determining the similarity between two distributions, we took the logarithm of these frequencies. That is to say, if an element took a value of r then this was replaced by $\log(r+1)$. Finally, for a given target word, its contextual representation was fixed as the concatenation of its preceding and succeeding frequency distributions (the resulting vector representation was thus 40,000 elements long). Pearson's correlation coefficient was applied to the distributions associated with two words to determine the degree to which they were similar.

In order to give some idea of the properties of this similarity measure, Table 1 shows the nearest neighbors of different inflections of the verb 'walk', with vectors derived from the BNC. As can be seen, the measure seems to exhibit the required properties in this context. Not only is it sensitive to the syntactic category of the target word, but the nearest neighbors, subject to this constraint, also seem to be semantically similar to the target word.

Decay Functions.

Once the nearest neighbors of a word have been identified, the predictive distributions associated with these words needs to be summed across in order to result in a neighboring distribution

	walk	**walks**	**walked**	**walking**
1	stay	walk	sat	putting
2	wait	rides	watched	talking
3	pick	smiles	waited	getting
4	sit	watches	stepped	doing
5	climb	drinks	laughed	sleeping
6	throw	dreams	talked	playing
7	talk	outdoor	tried	sitting
8	eat	lectures	stayed	eating
9	swim	laughs	asked	drinking
10	turn	poems	listened	wandering

Table 1: The nearest neighbors of different inflections of the verb 'walk'. The cue words are shown in bold font, and the nearest neighbors for these words are shown, in order, in the columns below them. The similarity measure appears to be sensitive to the specific inflection of the cue verb.

with which the directly observed bigram frequencies can be supplemented. We decided to explore three basic ways in which information from neighboring distributions can be weighed when they are being summed. All of these methods rely on a single parameter, p_1, to determine the precise form of their weighting scheme. The first of these is a power-decay curve. If u_i is used to denote the weight, between 0 and 1, which the ith nearest distribution receives, then for power-weighting

$$u_i \equiv (i+1)^{-p_1}, \text{ where } p_1 > 0 \tag{34}$$

The second type of decay we decided to explore was exponential weighting

$$u_i \equiv e^{-p_1 i}, \text{ where } p_1 > 0 \tag{35}$$

And the final type of decay was a simple k nearest neighbors approach, in which the first p_1 neighbors receive a weight of 1, and everything else receives a weight of 0:

$$u_i \equiv \begin{cases} 1 & \text{where } i \leq p_1 \\ 0 & \text{where } i > p_1 \end{cases} \tag{36}$$

These three schemes offer fairly diverse approaches to weighting, and hence might be able to provide useful information about the degree to which it is beneficial to generalize across similar linguistic items. For example, the power-based decay curve is long-tailed, meaning that it assigns non-negligible weights to distributions that are only remote neighbors of the distribution to be smoothed; on the other hand, the exponential distribution is short-tailed, meaning that it only assigns weight to a relatively few nearest neighbors. If there are performance differences between these three weighting schemes, then this could offer useful insight into the degree to which one should optimally generalize in language learning.

The Neighboring Distribution.

Once the nearest neighbors of a word and a decay-function have been set, the next step is to sum over the succeeding distributions of these words in order to arrive at a 'neighboring distribution' which can be used to supplement the directly observed bigram distribution. For convenience, we assume that the predictive distributions associated with the nearest neighbors are concatenated to form a matrix K in which k_{ij} represents the frequency with which the jth context word occurred after the ith nearest neighbor of the target word in question (the rows of K represent the neighboring distributions, starting with the nearest neighbor and moving to more and more distant neighbors; note that the target word itself is excluded in this representation even though, strictly speaking, a word is almost always its own nearest neighbor; this is done to avoid double-counting the directly observed bigram frequencies associated with w_{i-1}) . In this case, we can define the neighboring distribution as follows:

$$m_j \equiv \sum_i w_i \frac{k_{ij}}{\sum_j k_{ij}} \tag{37}$$

Note that even after summing across the nearest-neighbors, there is still no guarantee that the entries of the vector m will all be non-zero (K could have all zeros in a given column). Therefore, the possibility of adding a 'covering' distribution to the neighboring distribution is included in our framework (where the covering distribution could be the unigram distribution, for example, which is almost always entirely non-zero; we denote the *i*th element of this distribution c_i). The covering distribution is intended to reflect general expectations about language where no specific information, in the form of a similarity-based inference, is available. The covered neighboring distribution is defined as follows, where $0 \leq p_2 \leq 1$ is a free parameter controlling the relative weight of the covering distribution when compared to the neighboring distribution defined in Equation 37:

$$n_j \equiv \frac{m_j}{\sum_j m_j} + p_2 \frac{c_j}{\sum_j c_j} \tag{38}$$

These two processes result in the covered neighboring distribution which is used to supplement the directly observed conditional frequency distribution for the target word.

Mixing Distributions.

The final aspect of the similarity-based framework controls how the supplemental distribution, derived from the nearest neighbors of the target word and the covering distribution, is added to the directly observed bigram frequency distribution. If b_j denotes the jth element of the predictive bigram frequency distribution, and s_j denotes the jth element of the smoothed distribution generated by the similarity-based method, then this element is defined:

$$s_j \equiv \frac{b_j}{\sum_j b_j}(1 - (b_j + 1)^{-p_3}) + p_4 \frac{m_j}{\sum_j m_j} \tag{39}$$

Here, $p_3 > 0$ introduces a discount ratio to the directly observed bigram frequencies. This discounting scheme is non-linear in frequency, and allows observations with smaller frequencies to

Parameter	Function
p_1	Controls generalization over nearest-neighbor distributions.
p_2	Controls weight of covering distribution.
p_3	Controls discounting ratios of directly observed frequencies.
p_4	Controls proportion of similarity-based distribution added to raw bigram distribution.

Table 2: The function of the free parameters in the similarity-based smoothing framework.

be discounted more than those with higher frequencies. This reflects the fact that observations with smaller frequencies may be less reliable than those with higher frequencies, and may hence need to be subject to a greater degree of smoothing. This idea is consistent with the predictions of Good-Turing smoothing, and is analogous to the discounting schemes implemented in the Katz, Absolute Discounting, Kneser-Ney and Modified Kneser-Ney smoothing methods. The parameter $0 \leq p_4 \leq 1$ controls the degree to which smoothing occurs: when it is large, a large amount of smoothing takes place. To derive actual probability estimates from the s_js, all that is needed to do is normalize their final values so that they sum to 1. That is,

$$P_{SB}(w_i|w_{i-1}) \equiv \frac{s_i}{\sum_j s_j} \tag{40}$$

Table 2 describes the four free parameters of the similarity-based framework we have presented, and briefly states their function in the framework.

Measuring Smoothing Performance

The ultimate goal of a smoothing model is to allow more reliable estimates of the probability of linguistic events to be reached. The standard approach taken to evaluate the quality of a language model (derived from a particular corpus and a particular smoothing implementation) is to get the model to determine the probability of an unseen test corpus, using Equation 3. The rationale underlying this approach is that because the test corpus is distinct from the corpus on which the language model was trained, it will contain many ngrams that never occurred in the training data (e.g., Essen and Steinbiss, 1992; Brown et al. 1992). Therefore, if a smoothing method successfully manages to assign appropriate probabilities to these previously unseen ngrams, it will assign the test corpus a higher probability than a less successful method. A good smoothing method should assign relatively high probabilities to *all* transitions in a new sample of language – both those observed and unobserved in the training corpus – and as a consequence should end up assigning a high probability to the whole test corpus. Note also that because there is a finite amount of probability mass to allot (all probabilities conditioned on the same history must sum to 1), no model can trivially succeed in this task by maximizing all its probabilities: if one probability is increased then others must be reduced in order to keep the sum of probabilities equal to 1. Hence,

success in this task is attained when the estimated probabilities according to the language model equal the probabilities of events in the sample.

To ensure that the performance of a smoothing method is not simply due to an idiosyncrasy of the test corpus, the test corpus must be of a sufficient size to ensure statistical reliability (Charniak, 1993). A direct consequence of this is that the estimated probability of a test corpus, $P_n(T)$, typically becomes extremely small because it is defined as the product of the probability of each transition in the corpus, and any one of these probabilities individually will tend to be relatively small. Therefore, to avoid underflow problems in evaluation, the *cross-entropy* of the model, written $H(T)$, on the test corpus is calculated. The cross-entropy between model and corpus is actually a simple function of the probability of the corpus according to the model, but because it is defined in terms of logarithms the problem of underflow is avoided. The per-word cross-entropy of the model on a data-set is written

$$H(T) \equiv -\frac{1}{|T|} \log_2 P_n(T) \tag{41}$$

where

$$\log_2 P_n(T) = \sum_{i=1}^{N} \log_2 P(w_i | w_{i-n+1} \cdots w_{i-1}) \tag{42}$$

The cross-entropy can be interpreted as the number of bits it would take to encode the sequence T using the language model being tested. Therefore, the smaller this quantity, the better the model is doing (and, correspondingly, the larger $P(T)$ will be). Another common measure of performance is the *perplexity* of the model on the data-set, which is defined:

$$X(T) \equiv 2^{H(T)} \tag{43}$$

This quantity can be interpreted as measuring, in effect, the number of equiprobable options the model is choosing between when it tries to predict the next word in the test corpus. Clearly, smaller cross entropies and perplexities indicate better language model performance. Although the exact perplexity or cross-entropy one would expect on a bigram modeling task varies considerably depending on the nature of the test corpus and its relationship to the training data, typical perplexities range from 50 to almost 1000 depending on the exact nature of the test corpus T, how similar it is to the training corpus, and how much training data was provided (Chen and Goodman, 1998, p.7).

In the existing literature there are remarkably few studies which have sought to systematically examine the relative performance of a comprehensive suite of smoothing methods – such as the set of methods we have reviewed above – so that conclusions about the relative effectiveness of the various methods can be drawn. More commonly, a researcher or group of researchers will propose a novel smoothing technique and compare it to only one or two other smoothing methods (see, for example, Gale and Church, 1991, 1994; Jelinek and Mercer, 1980; Katz, 1987; Kneser and Ney, 1995). This has made it difficult to assess the relative performance of smoothing methods.

The notable exception to this pattern is the empirical work of Chen and Goodman (1996; 1998). Chen and Goodman set out to implement a wide range of techniques for smoothing, including almost all of the methods described above, and to apply them to a range of corpora, over a range of corpus sizes, and to both bigram and trigram models, in order to characterize the contexts in which each method can be expected to perform well. They also used the Wall Street Journal, the Brown, the North American Business, the Broadcast News and the Switchboard corpora in order to ensure that any trends were not due to properties of the corpus used, but were likely to generalize. In general, what they found was that the best performing models were the Modified Kneser-Ney and the Kneser-Ney models, lending support to the idea that the unique distribution these techniques rely upon really does capture something significant about the structure of language. As a result of this comprehensive study, it is fair to conclude that these techniques constitute the current state-of-the-art when it comes to language modeling.

Simulation 1

Simulation 1 was designed to assess the relative effectiveness of the various smoothing methods we have reviewed – including the similarity-based framework – at estimating transitional probabilities in language. In order to do this we randomly selected 200 words spread over a wide range of frequency bands, and used these to define the conditional probability distributions we would attempt to estimate: for each word w in the set of sample words we derived an estimate of the distribution over words following w. Our goal was to use varying amounts of training data, and to examine how close the conditional probability distributions predicted by each smoothing technique were to a corresponding *Gold Standard* distribution. Obviously the notion of the 'real' or 'Gold Standard' probability of an event in language is a problematic one – under the standard assumptions of sampling theory an empirically derived probability estimate only reaches asymptote 'in the limit' or, in other words, after an infinite amount of data has been observed. We therefore defined our Gold Standard distributions on a sample of language that was large relative to the training data available to the smoothing methods. Although these Gold Standard distributions will themselves be subject to data-sparsity to some degree, and cannot therefore be taken as ideal distributions which should be perfectly converged upon by a smoothing technique, they will nevertheless contain many occurrences never observed in the training data available to the smoothing methods, and hence provide a good test of the ability of the smoothing techniques to generalize from observed linguistic events in an appropriate way. The advantage of using this methodology over the more common language modeling approach in this instance is two-fold: many sets of parameter values for the smoothing methods can be explored, as the distribution associated with each sample word can be estimated in a fairly efficient fashion compared to the cost involved in estimating all the transitional probabilities in an extensive test corpus; and also the properties of the selected distributions, such as their frequency, can be readily controlled, which is not as easily accomplished with a randomly sampled test corpus.

Corpus

In order to provide the data with which to train and test the various probability estimation procedures we have reviewed, we decided to use the British National Corpus (BNC; Burnard, 2000; Burnage and Dunlop, 1992). The BNC is a corpus consisting of approximately 100 million words of both written (90%) and transcribed spoken (10%) text. The BNC was deliberately designed in order to include linguistic samples from a wide range of contexts, including different types of imaginative and informative texts, as well as recorded and transcribed conversations. In total, the corpus consists of 4054 different samples, none of which exceeds a length of 45,000 words. We decided to use such a diverse corpus for a number of reasons. We believed that a diverse corpus would make it less likely that a particular smoothing method would succeed simply because of idiosyncrasies of the corpus, thus making it more likely that the performance of the smoothing techniques would generalize to new and different data more consistently. We also believed that a diverse corpus would, if anything, provide a sterner test of generalization than a more homogeneous corpus: generalizing from distinct samples is much harder as there are likely to be a greater number of unobserved events between training and test samples. This should result in the relative performance of the smoothing methods being pulled apart more than if a more homogeneous corpus had been used.

Tokenization

Before statistical information can be extracted from a corpus, it is necessary to pre-process the data in the corpus, and to define exactly what is going to count as a token (word) of text. The BNC contains much information over and above its actual text, in the form of SGML tags that have been added by human-coders. We stripped this information from the corpus before we used it, as it was important that the models we were testing relied on no hand-coded information. This is because we are interested in exploring the relevance of similarity-based information to human language learners, and do not wish to assume that they have access to any prior knowledge other than the identity of the words they are exposed to and their statistical properties. We performed tokenization by using a simple tokenization algorithm which (i) removed SGML mark-up, (ii) split tokens on white-space, (iii) treated punctuation marks as separate tokens, and (iv) included some simple rules to separate possessive constructions and maintain acronyms as coherent tokens.[5] As a result of this tokenization procedure, the total corpus we were left with consisted of 115,011,693 tokens, and 551,184 distinct word types (the number of tokens was inflated from approximately 100 to 115 million largely because punctuation and possessive markers were treated as separate tokens according to our tokenization algorithm).

Methodology

We used the first 70% (approximately 80M words) of the BNC in order to define the Gold Standard distributions for the 200 words that we sampled; these were defined using straightforward

[5]For example, the string 'went.' would be transformed into two separate tokens, 'went' and '.', whereas the string 'A.B.C.' would be preserved as a single token because it would be recognized as an acronym.

maximum likelihood estimation. The remaining 30% of the data (approximately 35M words) was used as the training data for the smoothing methods. We restricted our vocabulary to the 19,999 most frequent word types in the BNC. This meant that the distributions we estimated consisted of 20,000 elements, with the first 19,999 elements corresponding to the most frequent word types, and the 20,000th element corresponding to a 'catch all' event (representing the occurrence of a token not included in the first 19,999 types). In this way, the distributions we were concerned with could be represented as vectors of 20,000 elements which are guaranteed to be genuine probability distributions summing to 1.

We examined the performance of each smoothing method at 6 points throughout the training data (each additional section thus constituted approximately another 5.7M tokens of training data). At each point in the training data we used a simulated annealing algorithm to estimate the free parameters of each method requiring this (Galassi et al., 2005). The parameters were set by minimizing the distance between the estimated probability distribution for each of the 200 words and the corresponding Gold Standard distribution estimated from the first 50% of the Gold Standard section of the corpus. We decided to optimize the free parameters of the models in this fashion because we wished to measure the absolute quality of the information that each smoothing method relied upon in a way that was independent of the quality of the estimated values for the free parameters; we nevertheless chose to withhold half of the data contained in the test corpus in order to avoid over-fitting artifacts. For each smoothing method, at each of the selected points in the training corpus, we therefore recorded the distance of the estimated distribution from the corresponding Gold Standard distribution. The distance metrics we used to compare the estimated distributions to the Gold Standard ones are discussed below.

One potential concern with our use of Gold Standard distributions in order to measure the success of a smoothing method is that the Gold Standard distributions could themselves be so afflicted by data-sparsity – because they are estimated using the maximum likelihood method – that they are essentially arbitrary, and hence getting close to them is not a meaningful measure of the success of a smoothing technique. In order to mitigate this objection somewhat we defined the Gold Standard distributions using more than twice the amount of data that the smoothing methods had at their disposal. In addition, we were careful to ensure that none of our sampled words had extremely low frequencies, as we show below, which would certainly compound data-sparsity problems. In fact this is a general problem with the evaluation of smoothing methods, and not one particular to our Gold Standard methodology. For example, the estimates of cross-entropy which one arrives at in the standard language modeling paradigm are only as good, or as representative, as the test sample itself is, and given that most test samples consist of only tens of thousands of words, any claim about representativeness is hard to sustain (see Charniak, 1993). In any event, we believe that such concerns are deflated in the present instance for the following reason: should the information contained in the gold standard distributions be so heavily degraded by data-sparsity so as to be rendered useless, we would expect to observe no differences in the performance of the smoothing techniques in terms of their ability to approximate these distributions, because the methods would, in effect, simply be chasing noise. As we will see, we obtain clear differences between the levels of performance of the smoothing methods and, moreover, the relative ordering

of the models in terms of their performance is in general aggreement with previous studies (e.g., Chen and Goodman, 1996,1998), which suggests that the Gold Standard vectors do indeed contain meaningful information about the transitional probabilities of events in English.

Sample of Words

We wanted the sample of words that were used to define the distributions to be estimated in Simulation 1 to include a wide range of frequencies so that the performance of the various smoothing methods could be evaluated across these ranges. Accordingly, we randomly sampled the words at equal intervals in logarithmic space according to their frequency in the BNC. This produced a sample of words which ranged from having millions down to hundreds of occurrences in the BNC. Figure 2 shows the corpus frequency of the 200 sampled words plotted against their rank frequency in the sample, showing the wide coverage of corpus frequencies explored, and the equal representation that each order of magnitude of frequency received.

We wanted to ensure that the occurrences of the sampled words occurred at relatively regular intervals in the corpus, otherwise a situation could have arisen in which all the occurrences of a word occurred solely in the training or the test portions of the data, which could make the performance of the smoothing methods vary in unpredictable ways.[6] Figure 3 therefore shows the log frequency of each sampled word at various points throughout the training corpus. Note that the frequency profiles for the words increase at a more or less constant rate as a function of the amount of data used (although this regularity is somewhat less reliable for the low frequency items, as is to be expected because of the greater sampling error in these cases). Another way to assess this issue is to look at the correlation between the frequency of each word in the training and test portions of the corpus. A scatterplot of this data is shown in Figure 4. As can be seen, there is an extremely high correlation between this frequency data ($R^2 = 0.97$, with one notable outlier) which shows that the two parts of the corpus are representative of one another in terms of the relative frequency of the sample words. This plot also shows that almost all of the lowest-frequency items nevertheless occur tens of times in the training corpus, meaning that there will be at least some item-specific information for the smoothing techniques to exploit. It also shows that the lowest frequency items occur at least hundreds of times in the test corpus used to define the Gold Standard distributions, which suggests that the Gold Standard distributions are insulated from the most serious consequences of data-sparsity.

Performance Measures

We used two distinct types of distance metric in order to measure the performance of the smoothing methods at estimating the Gold Standard probabilities in Simulation 1. The first measure used was the Kullback-Leibler distance metric (KL distance; Kullback and Leibler, 1951). The Kullback-Leibler measure is motivated from an information-theoretic standpoint which aligns it with the fundamental goals of language modeling, and is defined for two discrete probability

[6] Baayen (2001) devotes some attention to characterizing the degree to which words occur in 'bursts', in other words the amount by which the rate of occurrence of word types varies throughout a corpus.

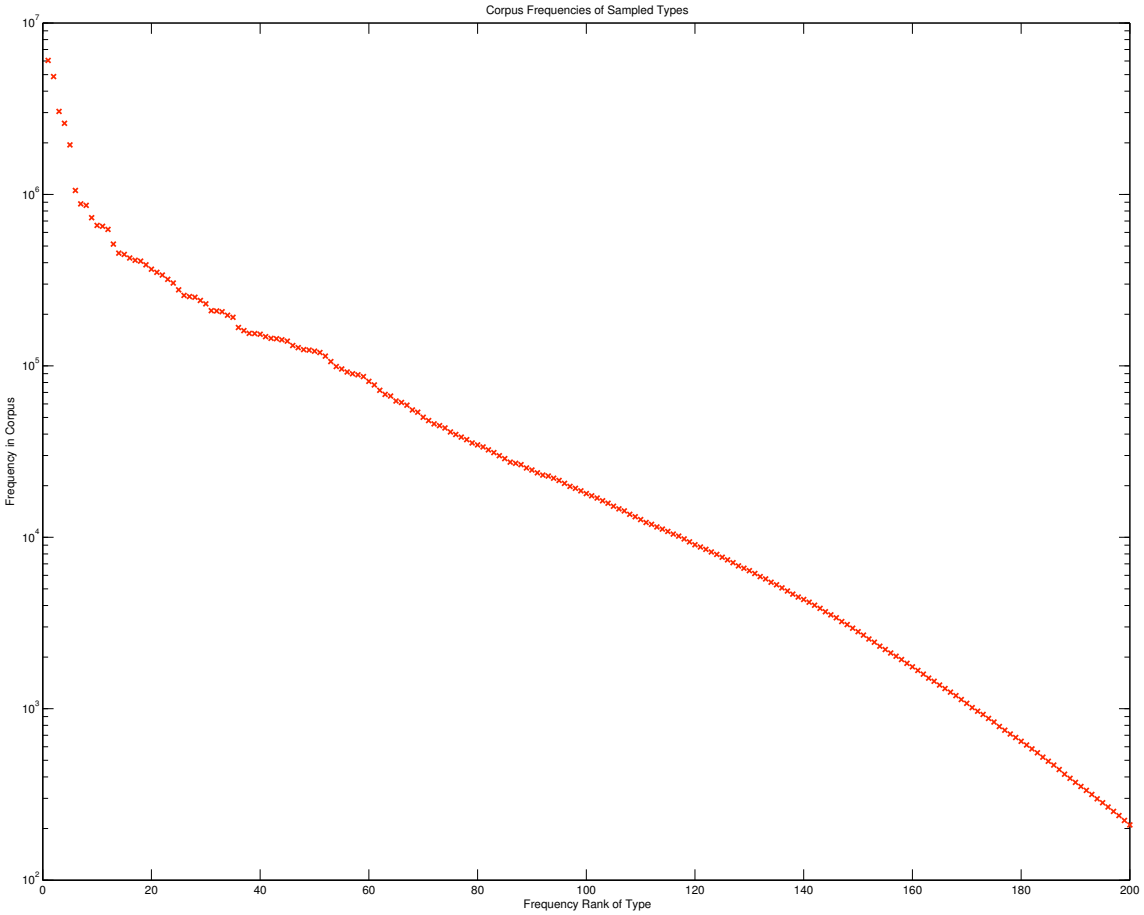

Figure 2. The log frequency of the 200 sample words from Simulation 1 in the BNC, plotted against their rank frequency in the sample. Note the coverage of a wide range of frequencies.

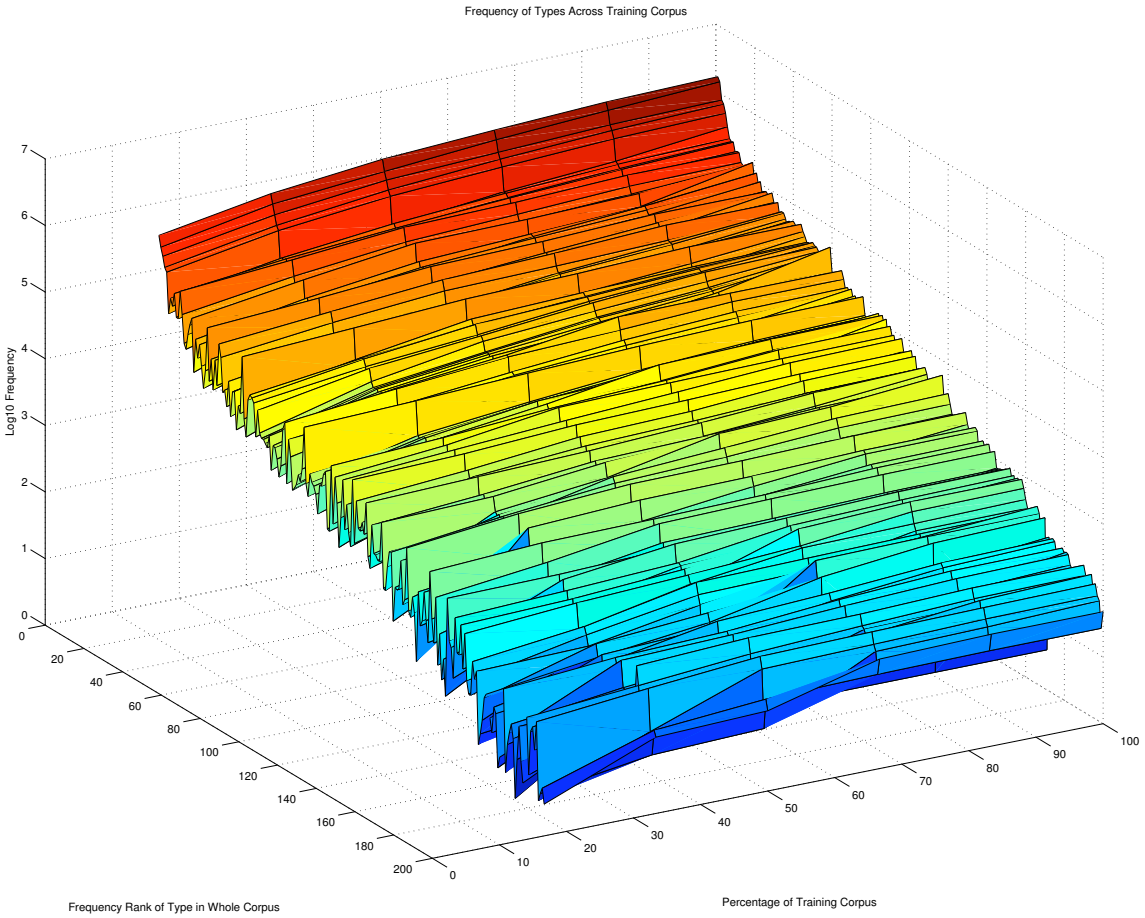

Figure 3. The log frequency of each of the 200 sample words plotted against the percentage of training corpus used. Note the more or less smooth increase in frequency for the words, showing that they are distributed fairly homogeneously over the corpus intervals used. Note, however, that this is less true for the very low-frequency items, as is to be expected due to the smaller number of occurrences for these items.

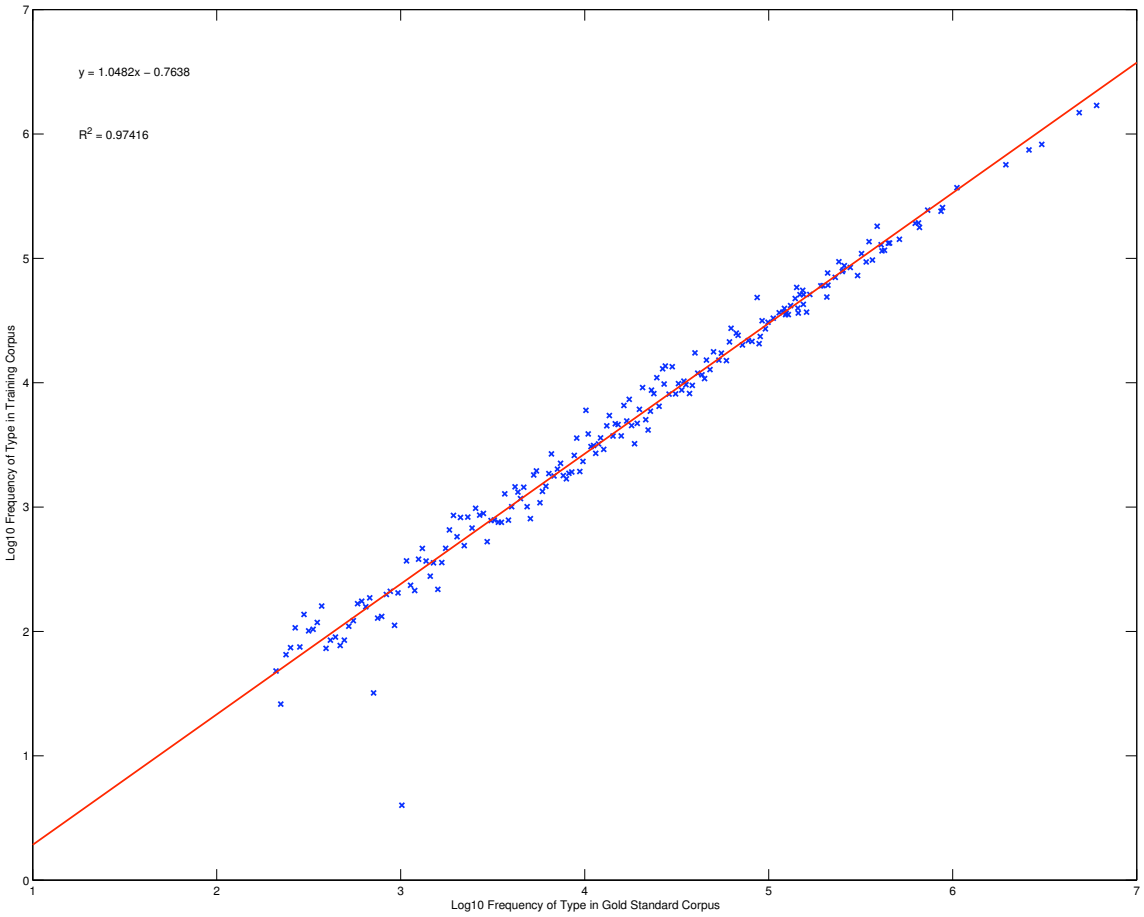

Figure 4. A scatterplot of the log frequency of each item in the Gold Standard Corpus (x-axis) against its log frequency in the Training Corpus (y-axis). The strong linear correlation indicates that the sampled 200 words are equally well represented by the two corpus samples.

distributions P and Q as follows:

$$D(P||Q) \equiv \sum_i p_i \log_2 \left(\frac{p_i}{q_i} \right) \tag{44}$$

If Q is assumed to represent an estimate of a 'real' probability distribution P – in this case, one of the Gold Standard distributions – then the KL distance of Q from P measures, in bits, the amount of information that is lost by the assumption that P is really Q. The measure is thus not symmetric – $D(P||Q)$ is not equal to $D(Q||P)$ in general. This is a sensible measure to use in the present context for a number of reasons. First, it is closely related to the cross-entropy measure used in standard language modeling tasks; we would therefore expect the relative performance of the smoothing methods in Simulation 1 as measured by the Kullback-Leibler distance metric to predict performance on a conventional language modeling task. KL distance can also be interpreted as being especially sensitive to what we could term the 'recall' of a smoothing method (to borrow terminology from the field of information retrieval). In the present context a smoothing method with high recall would be one which fills in all the 0 frequencies that need to be filled in in a directly observed bigram frequency distribution (we can think of these events as having been successfully recalled). KL distance is especially sensitive to failures in recall: if the real distribution, P, contains a non-zero value in a particular location but the estimated distribution contains a zero value in this position, then the KL distance becomes infinite (the denominator of the fraction in Equation 44 becomes 0). In other words, a failure of recall is infinitely punished under the KL metric.[7]

The second family of distance measure we decided to explore was the Minkowski family. This family of measures is defined as follows:

$$L_r(P,Q) = \left(\sum_i |p_i - q_i|^r \right)^{\frac{1}{r}} \tag{45}$$

where r determines the norm of the measure. In particular, we focused on the familiar Manhattan ($r = 1$) and Euclidean ($r = 2$) metrics in order to evaluate our smoothing metrics. Whereas one can think of KL distance as being sensitive to recall, the Minkowski metrics are better thought of as being sensitive to 'precision'. If an estimated distribution includes a 0 frequency when the Gold Standard distribution contains a non-zero value in the corresponding location, the Minkowski metrics do not penalize this occurrence as a special case. Instead, they are sensitive to the arithmetic

[7] This actually leads to a problem with KL distance: when a bigram probability fails to be recalled, the measure becomes infinite, which does not allow the performance of the various measures to be compared (Lee, 1999). While this is consistent with the tenets of information-theory – we have, in a certain sense, done infinitely badly if we have estimated an event to be impossible when it is, in fact, not – this is unhelpful when it comes to the issue of model comparison. Therefore, in our simulations, any element in Q (the estimated distribution) which would cause such a situation to arise was replaced by an extremely small probability (10^{-14} to be exact, which is smaller than any empirically derived probability estimate could be in our simulations). This results in a large punishment term being levied on any model guilty of such a sin of omission, while still allowing the relative performance of the various models to be assessed in a meaningful fashion.

difference between the estimated and 'actual' probabilities, considered across the whole distribution.[8] Our intention was that by using a variety of distance measures, that we would be able to assess the relative performance of the smoothing techniques based on a number of different criteria, and thus perhaps gain a more representative picture of where each smoothing algorithm was succeeding and failing.

Distributional Sparsity

Although we know that data-sparsity for bigrams will be a problem even given large amounts of training data, exactly how severe is this after tens, or even hundreds of millions of words of training data? Some insight into this issue can be gained by examining the frequency of frequency distribution for bigrams (otherwise known as the bigram frequency spectrum; Baayen, 2001) – that is, we can examine the number of bigram types which occurred once, twice, and so on, in our sample. This distribution is shown in Table 3, and is plotted logarithmically in Figure 5.

We can use this data to estimate the number of unobserved bigrams given our sample (i.e. the number of bigrams with a frequency of 0).[9] Because we restricted ourselves to a vocabulary of 20,000 items, there are a total of $20,000^2 = 400$ million possible bigrams for us to consider. If we assume that the logarithm of the number of bigrams with a given frequency decreases linearly with the logarithm of the frequency under consideration – following Gale (1995) – then we can fit a regression model to this data. This analysis showed that $\log_{10}(\text{Frequency of frequency}) = 6.4743 - 1.7338 \log_{10}(\text{Frequency})$ ($R^2 = 0.999$, $p \approx 0$). According to this linear fit, the last frequency which more than 0 bigrams will exhibit is 5287. Therefore, we can sum over the linear predictions of the number of bigrams for these values in order to get an estimate of the total number of observed bigrams in our sample. Doing this yields a total number of observed frequencies of 5.8992×10^6. Subtracting this from the total number of possible bigrams suggests that there are approximately 394 million bigrams with a frequency of 0 in our sample. Of course, the proportion of these that are 'illegitimate' bigrams – ones that we would not expect to find in any language sample, no matter how large – and the proportion that were unobserved simply because of data-sparsity cannot be estimated from this data.

A number of interesting observations can be made about this frequency of frequency data. Perhaps the most salient facts to observe are that the vast majority of bigrams – 98.5% of them – were never observed at all, indicating that data-sparsity could be a potentially serious issue even given a sizable training corpus, and that of the observed bigrams, about 59% of them were observed only once. The twin aspects of data-sparsity are present here, then, even with a relatively large training corpus and even though we are only concerned with bigram statistics (and not higher-order

[8] It is important that the distributions in question have the same magnitude when they are compared, otherwise the magnitude of each distribution would be a confounding factor in measurement. For the 1-norm this simply means that the distributions must sum to 1, as was the case with KL distance. However, for the 2-norm, the vectors have to be linearly scaled so that $\sqrt{\sum_i p_i^2} = 1$.

[9] Although one could have acquired this figure directly, by subtracting the number of observed bigram types in the sample from the total number of possible bigrams given the vocabulary size, this would have required keeping track of the millions of distinct bigrams that were observed, which we did not record because of memory limitations.

Frequency	Frequency of Frequency	% of possible bigrams
0	394,000,000	98.5252% (Estimated)
1	3,479,453	0.8699%
2	901,750	0.2254%
3	426,623	0.1067%
4	255,721	0.0639%
5	172,938	0.0432%
6	127,040	0.0318%
7	96,996	0.0242%
8	76,573	0.0191%
9	62,562	0.0156%
10	52,781	0.0132%
11	44,863	0.0112%
12	38,764	0.0097%
13	33,854	0.0085%
14	30,163	0.0075%
15	26,817	0.0067%
16	23,905	0.0060%
17	21,670	0.0054%
18	19,697	0.0049%
19	17,953	0.0045%
20	16,384	0.0041%
21	15,142	0.0038%
22	14,089	0.0035%
23	13,019	0.0033%
24	12,072	0.0030%
25	11,186	0.0028%
26	10,516	0.0026%
27	9,639	0.0024%
28	9,528	0.0024%
29	8,906	0.0022%
30	8,286	0.0021%

Table 3: The frequency of frequency table for bigrams, as derived from the BNC.

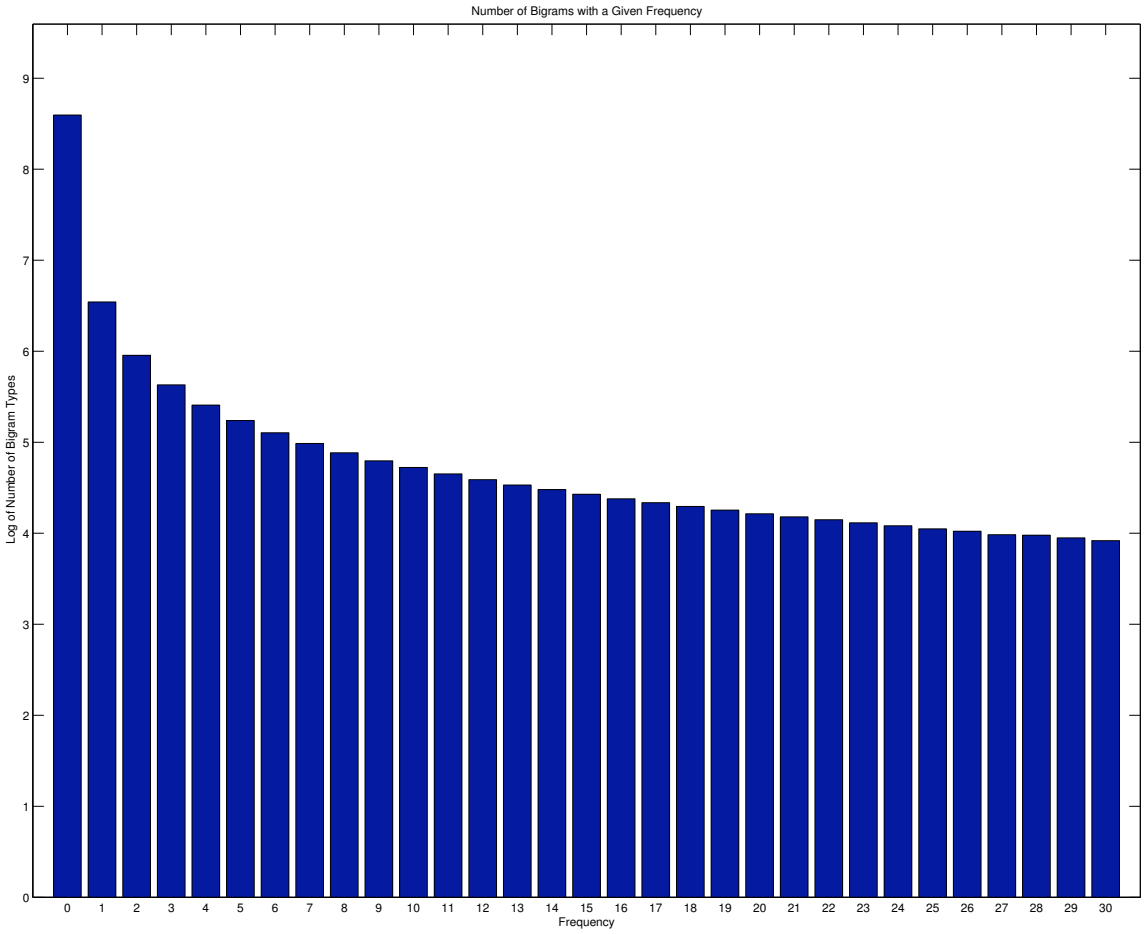

Figure 5. Log of the number of bigrams which occurred in the BNC with frequency 0-30. Note that the number of bigrams occurring 0 times in the corpus is 2 orders of magnitude greater than the next most frequent frequency. Note also the orderly decrease in the number of bigram types occurring with a given frequency, as frequency increases. The vast majority of the V^2 possible bigrams were never observed at all, indicating the potential magnitude of data-sparsity issues.

ngrams): the majority of possible bigrams were never observed – although, admittedly, many of these could be bigrams that we would not expect to observe no matter how large the training corpus used – and, of those that were observed, a considerable fraction were observed only once. How do these facts affect the conditional probability distributions that can be estimated using simple maximum likelihood?

In order to get a sense of the general properties of the conditional probability distributions we were attempting to estimate, and hence how much they might be affected by data-sparsity, we examined the empirical characteristics of these 200 distributions given varying amounts of training data. The first thing we did was to examine the percentage of entries in each distribution that had a frequency of 0, as a function of the training data used. This data is shown plotted in Figure 6. The percentage of elements with an observed frequency of 0 in the distributions is plotted separately for words falling in each quartile of the sample words by frequency. It can be seen, therefore, that only the distributions associated with the 50 most frequent words in the sample come to have an appreciable proportion of entries that have non-zero frequencies – and even then, only around 25% of the possible entries, at most, are non-zero. This is a significant fact when it comes to thinking about data-sparsity: even the most frequent words tend to have associated frequency distributions in which the majority of items have 0 frequencies. In addition, we can estimate how much training data might be required before these curves asymptote at their terminal levels – indicating that all word types that could legitimately follow the target word in question have been observed. However, the gradient – strictly speaking, the second differential – of these curves suggests that these asymptotes would only be reached after training data many times the size of that used in the present simulation was utilized. This suggests that, at least given the available training data, the raw frequency distributions are far from completely estimated, and that there remain many unobserved transitions in them which need to be estimated through smoothing.

A complementary analysis to the examination of the 0 frequencies in the sampled bigram frequency distributions is to ignore the zero-frequency observations in these distributions and examine the mean value of the non-zero frequencies in these distributions. This data is plotted in Figure 7. Here the highest frequency words have distributions with a mean non-zero frequency of around 30, whereas the 50 lowest-frequency words have distributions with a mean non-zero frequency of somewhere between 3 and 4. Again, we can see that if the observed bigram frequencies have, on average, a value of 3 or 4, that the probabilities associated with them are likely to be subject to very high variance, and hence be unreliable. Once again we are led to the conclusion that data-sparsity seems to be a serious problem for bigram statistics, even after around 35M words of training data (only slightly less than they typical 12-year-old is estimated to have experienced; Landauer and Dumais, 1997).

Coverage

As previously described, we restricted our attention to the most frequent 19,999 words in the BNC when estimating the conditional probability distributions for the sampled words. An important issue, then, is to consider the amount of coverage – the percentage of word types that fall within this set of words – that the lexicon we have employed achieves. Clearly the greatest

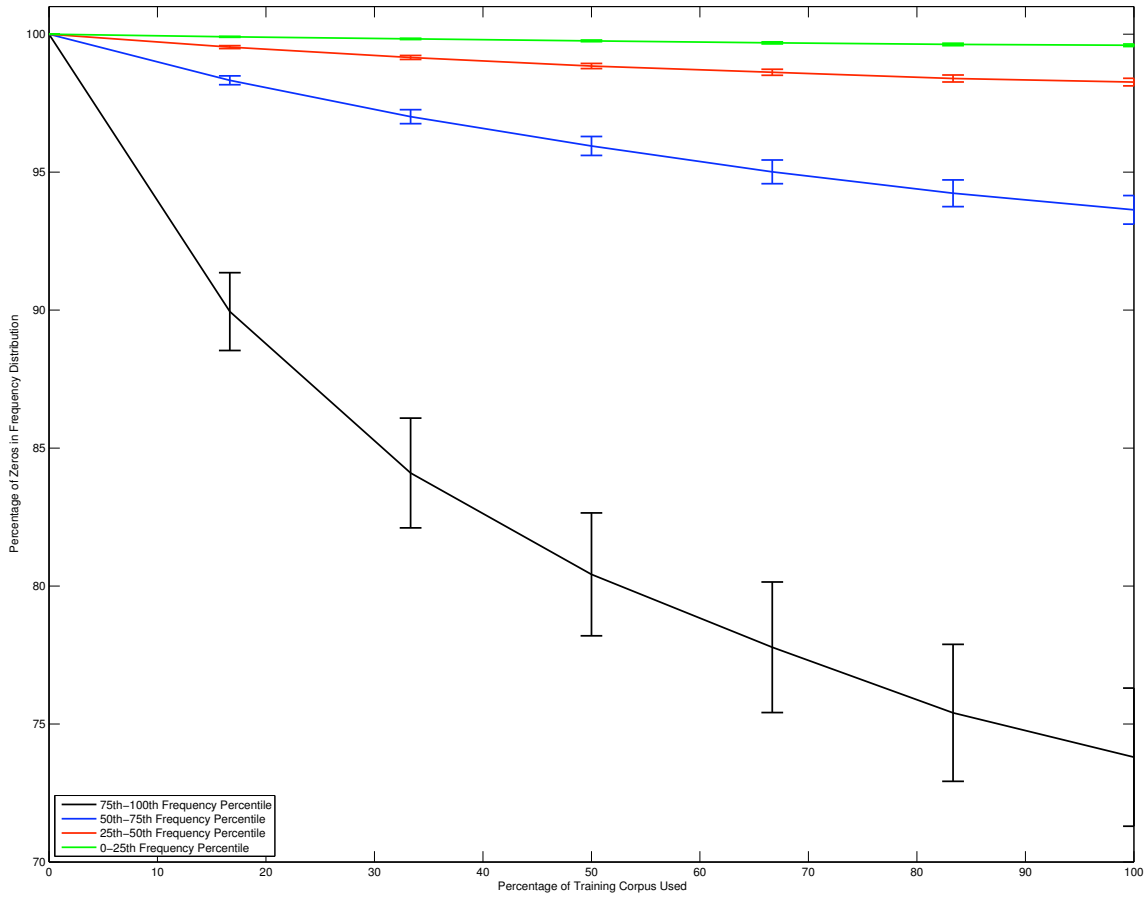

Figure 6. The mean percentage of zero frequencies in the frequency distribution for each quartile of the sampled words (y-axis) plotted against the percentage of the training corpus used (x-axis). Note that even the frequency distributions for the high frequency words consist of approximately 75% zero entries at the end of the training corpus, indicating the general sparsity of the conditional probability distributions.

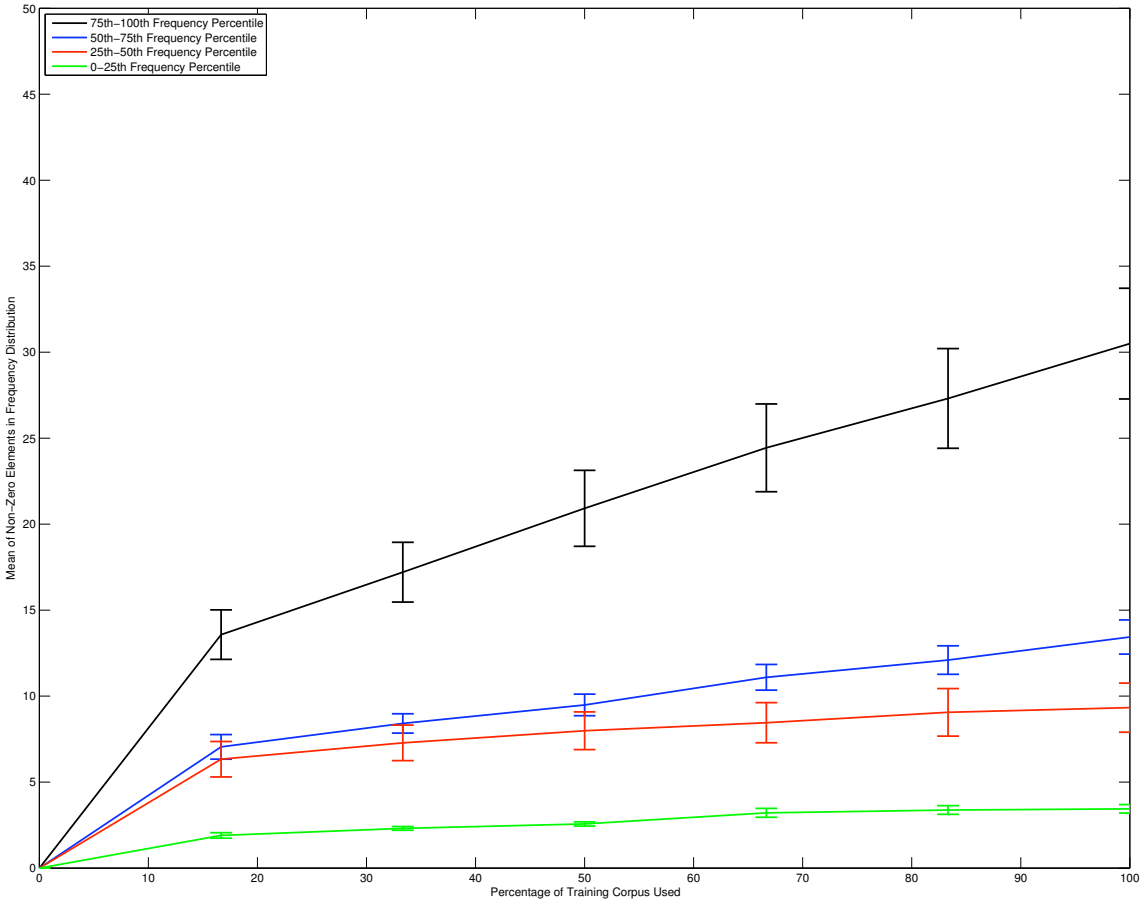

Figure 7. The mean of the non-zero elements in the frequency distributions plotted against the percentage of training corpus used, and stratified into the 4 quartiles of rank frequency. Note that the lower quartile of the corpus has effectively negligible values in the frequency distribution on average.

possible coverage is desirable, as it is only in this case that predictions about specific words will be made by the smoothing models (otherwise the probability estimate will default to the probability of the 'catch all' event). In order to examine the sort of coverage attained with this size of language model – which uses, for example, the same vocabulary size as used by Dagan, Lee and Pereira (1999) – we counted the proportion of events in our distributions which fell into the final, 'catch all', element. This data is plotted in Figure 8. As can be seen, this proportion was more or less constant among the 200 distributions of interest as a function of training data, staying constant at a level of around 0.04. Therefore, in our predictions we attained a coverage of around 96% by using the first 19,999 most frequent words. This is an acceptable rate of coverage, which ensures that our language models are making specific predictions about upcoming events for the vast majority of word types.

Good-Turing Results

As previously discussed, the Good-Turing method provides a principled way of adjusting the empirically observed frequencies of bigrams in order to make them more closely reflect the expected frequency of the same item in a new sample, but taking into account the information contained in the first sample. The Good-Turing replacement frequencies which were estimated from the BNC are listed in Table 4, and are plotted on a logarithmic scale in Figure 9. In general, one can see that bigrams with a frequency of 0 in the BNC (i.e. unobserved bigrams) should be treated as though they had a frequency of 0.0088, indicating that they have a modest, albeit relatively small, chance of occurring despite the fact that they have yet to be observed. Note, in addition, that the frequencies of the observed bigrams (i.e. those with a frequency greater than 0) are discounted – for these frequencies, $r^* < r$ – in order to compensate for the frequency mass which is redistributed to the bigrams with a frequency of 0.

Models

Many of the existing smoothing models we have described include methods for setting the value of the parameters they use based on properties of the observed input. Where these methods existed for a model we used them to generate a set of predictions. We refer to this version of the model as its *fixed* implementation. In addition, for each model, irrespective of whether it has a *fixed* version (i.e. independent of whether a method for determining its parameter values has been specified), we implemented a *free* version of it, in which its free parameters were set according to an optimization process based on a simulated annealing algorithm (Galassi et al, 2005). Thus, the *free* version of a model is guaranteed to do at least as well as the corresponding *fixed* version, where one exists (we used a generous annealing schedule, and thus did not anticipate the optimization process being so hampered by local optima that it failed to find parameter values which performed at least as well as the fixed parameters).

We implemented three different varieties of similarity-based model for use in Simulation 1, each implementation using one of the possible decay-functions: power-decay, exponential-decay, and the k nearest neighbor decay. We did this in an attempt to see if there were significant performance differences between the various methods which might throw some light on the degree

Frequency (r)	Good-Turing Replacement Frequency (r^*)	Ratio (r^*/r)
0	0.0088	Inf
1	0.5183	0.5183
2	1.4193	0.7097
3	2.3976	0.7992
4	3.3814	0.8453
5	4.4076	8815
6	5.3446	0.8908
7	6.3156	0.9022
8	7.3532	0.9192
9	8.4366	0.9374
10	9.3498	0.9350
11	10.3686	0.9426
12	11.3534	0.9461
13	12.4736	0.9595
14	13.3360	0.9526
15	14.2626	0.9508
16	15.4106	0.9632
17	16.3611	0.9624
18	17.3177	0.9621
19	18.2521	0.9606
20	19.4081	0.9704
21	20.4701	0.9748
22	21.2533	0.9661
23	22.2542	0.9676
24	23.1652	0.9652
25	24.4427	0.9777
26	24.7483	0.9519
27	27.6776	1.0251
28	27.1068	0.9681
29	27.9115	0.9625

Table 4: The Good-Turing replacement frequencies as calculated from the BNC.

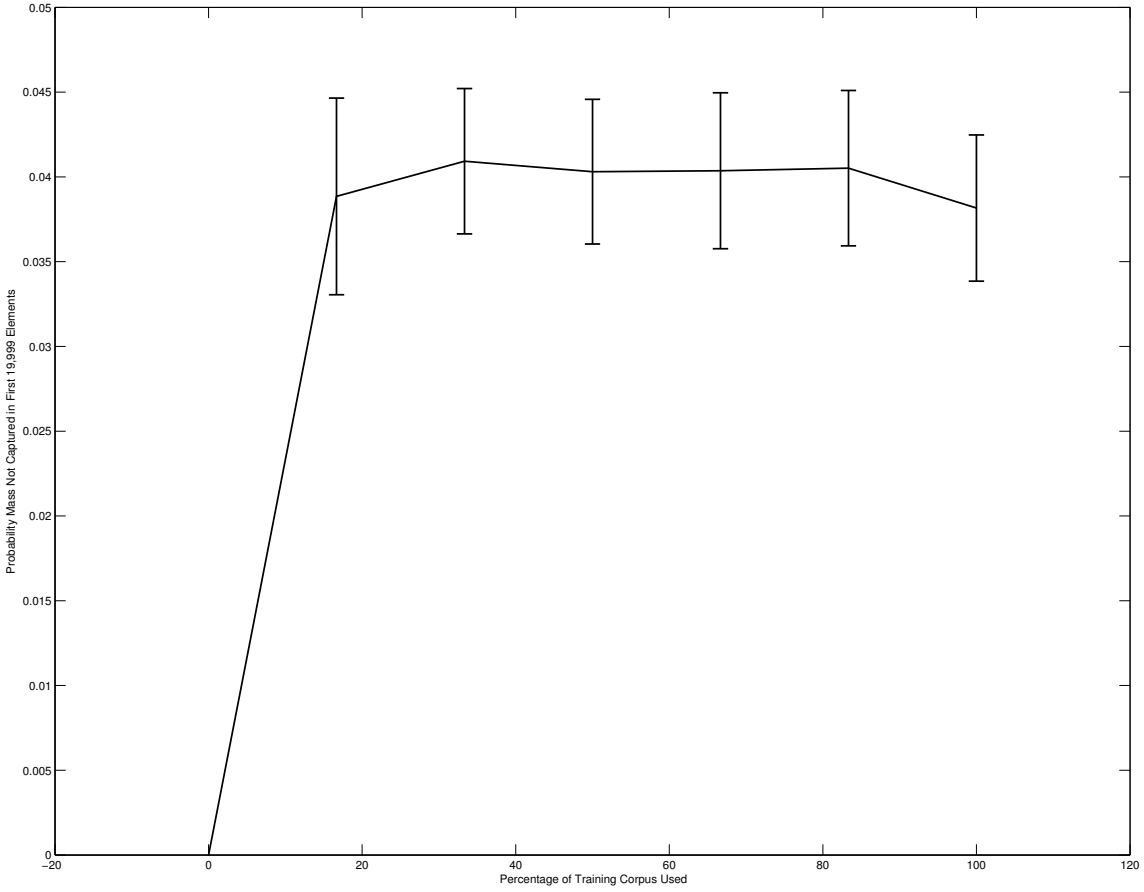

Figure 8. The proportion of word types falling within the 'catch all' event for the 200 sampled distributions, plotted as a function of the amount of training data used.

to which it is beneficial to generalize across neighboring distributions when estimating transitional probabilities. One caveat is in order with respect to the implementation of the power-decay and exponential-decay models. Depending on the estimated value of p_1, it is possible for the power- and exponential-decay models to assign non-negligible weights to even extremely remote 'neighboring' distributions. However, the computational complexity of the similarity-based framework we have outlined is $O(k)$, where k is the number of distributions that have to be summed over. Therefore, in order to minimize the (considerable) computational cost involved in running Simulation 1 we were forced to restrict the integration process to the 1000 nearest neighbors, even if the decay-weights derived from p_1 had not yet decayed to 0. This truncation in the sum over neighboring distributions is unlikely to have had a great impact on the performance of the similarity-based models in Simulation 1: a subsequent analysis of the p_1 values that were reached through simulated

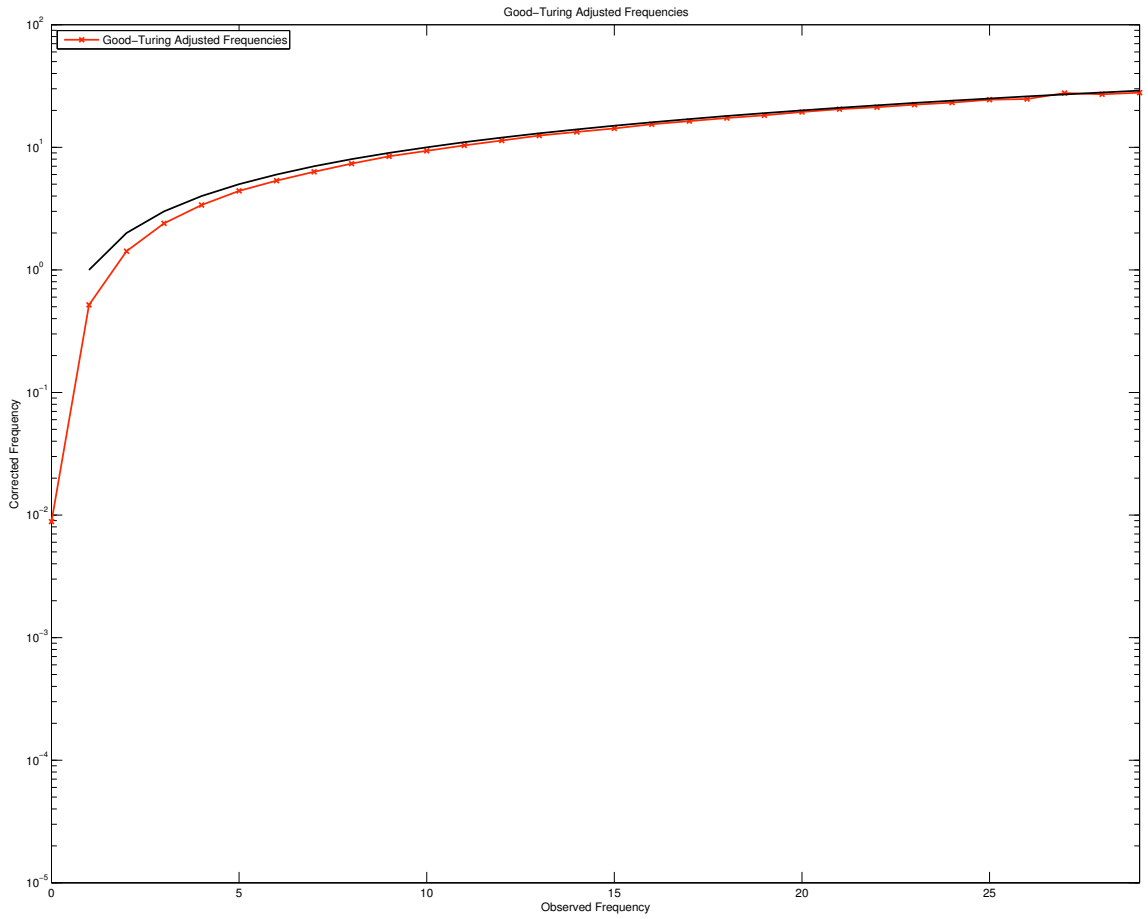

Figure 9. Good-Turing discounted frequencies plotted against raw frequencies. The Good-Turing frequencies are plotted against the empirically observed frequency counts, as derived from the entire BNC with a vocabulary size of 20,000 tokens. The Good-Turing method replaces 0 frequencies with 0.0088, and reduces all other frequencies by slightly reduced amounts in order to compensate for the probability mass assigned to the 0-frequency events.

annealing showed that in most cases the decay-weights would have reached 0 or thereabouts by the 1000th nearest neighbor. Note that this issue did not affect the k nearest neighbor algorithm: in no case did the value of p_1 estimated through simulated annealing for the k nearest neighbor algorithm approach a value of 1000.

Distances from Gold Standards

We now report the performance of the various smoothing methods in Simulation 1 by examining the distances of the distributions defined by these methods from the corresponding Gold Standard distributions. The mean distance of the distributions estimated by each smoothing method from the appropriate Gold Standard distributions, after all the training data has been utilized, is shown in Table 5. The best-performing methods for each distance metric are shown in bold; in each case, the estimation method which consistently performed best – i.e. got closest to the Gold Standard distributions – was one of the 3 similarity-based techniques which we proposed.

In order to examine the performance of the similarity-based techniques statistically we compared the performance of the similarity-based techniques to the next best-performing technique using all of the available training data. The similarity-based techniques were universally the best-performing methods: in each case they got significantly closer to the Gold Standard vectors on average than all the other smoothing techniques.

As measured by Kullback-Leibler distance, the next best-performing technique was the Modified Kneser-Ney Free method. A paired t-test was conducted on the distances attained by the Similarity-Based techniques and the Modified Kneser-Ney Free technique. Both the Power-decay ($t_{199} = 3.78$, $p = 0.0002$) and the Exponential decay ($t_{199} = 3.07$, $p = 0.0024$) versions of the Similarity-Based technique performed significantly better than the Modified Kneser-Ney Free technique. There was no significant difference between the Modified Kneser-Ney Free technique and the k-nearest-neighbor version of the similarity-based technique ($t_{199} = 0.92$, $p = 0.3577$). Under the Manhattan distance metric, the next best-performing technique was the Absolute Discounting Free method. All three of the similarity-based techniques performed significantly better than this technique: the Power-decay ($t_{199} = 8.48$, $p = 4.82 \times 10^{-15}$), the exponential-decay ($t_{199} = 8.83$, $p = 5.70 \times 10^{-16}$), and the k-nearest-neighbors version ($t_{199} = 9.29$, $p = 2.70 \times 10^{-17}$) all got significantly closer to the Gold Standard probability distributions than the Absolute Discounting Free method managed to. Under the Euclidean distance metric, the next-best performing technique was the Jelinek-Mercer method. Again, the similarity-based techniques, in each case, performed significantly better than this method: the Power-Decay ($t_{199} = 10.21$, $p = 6.10 \times 10^{-20}$), the Exponential-Decay ($t_{199} = 10.43$, $p = 1.33 \times 10^{-20}$), and the k-nearest-neighbors version ($t_{199} = 10.53$, $p = 7.05 \times 10^{-21}$) all performed better than the Jelinek-Mercer method.

A graphical summary of the performance of the various smoothing techniques, as evaluated by Kullback-Leibler distance, is shown in Figures 10, 11 and 12. Figure 10 shows the performance of the smoothing methods that were, in general, less successful at estimating the linguistic probabilities, including simple Maximum Likelihood estimation, the Uniform method, a variety of Good-Turing methods, the Add One and Add Delta methods, and Katz smoothing. One can gain an appreciation of the problems faced by maximum likelihood estimation by observing that it is

Smoothing Method	Kullback-Leibler	Manhattan	Euclidean
Maximum Likelihood	4.6805 (0.3430)	0.5710 (0.0199)	0.3308 (0.0170)
Uniform	8.0267 (0.1154)	1.8696 (0.0087)	1.3904 (0.0007)
Good-Turing	1.3557 (0.0966)	0.6940 (0.0301)	0.3287 (0.0170)
Simple Good-Turing (Global)	1.5656 (0.1061)	0.6236 (0.0244)	0.3340 (0.0173)
Simple Good-Turing (Tokenwise)	1.4555 (0.1174)	0.6223 (0.0230)	0.3385 (0.0181)
Plus One	3.1298 (0.1801)	1.3897 (0.0399)	0.5427 (0.0289)
Plus Delta	1.2159 (0.0914)	0.5710 (0.0199)	0.3307 (0.0170)
Katz	1.0600 (0.0800)	0.5831 (0.0199)	0.3281 (0.0170)
Katz (Kneser-Ney)	1.0124 (0.0765)	0.5909 (0.0206)	0.3278 (0.0170)
Simple Back-Off	1.1681 (0.0721)	0.7964 (0.0183)	0.3442 (0.0168)
Witten-Bell	1.3004 (0.1021)	0.5771 (0.0194)	0.3285 (0.0168)
Witten-Bell (Kneser-Ney)	1.1928 (0.0968)	0.5873 (0.0195)	0.3306 (0.0169)
Jelinek-Mercer	1.0201 (0.0704)	0.5657 (0.0189)	0.3091 (0.0144)
Jelinek-Mercer (Kneser-Ney)	0.9744 (0.0687)	0.5702 (0.0196)	0.3197 (0.0154)
Absolute Discounting Fixed	1.0866 (0.0786)	0.5649 (0.0193)	0.3223 (0.0166)
Absolute Discounting Free	1.0001 (0.0715)	0.5484 (0.0188)	0.3101 (0.0156)
Kneser-Ney Fixed	0.9989 (0.0745)	0.5879 (0.0206)	0.3267 (0.0169)
Kneser-Ney Free	0.9479 (0.0690)	0.5534 (0.0195)	0.3202 (0.0165)
Modified Kneser-Ney Fixed	1.0020 (0.0678)	0.5899 (0.0194)	0.3274 (0.0164)
Modified Kneser-Ney Free	0.9407 (0.0678)	0.5505 (0.0194)	0.3176 (0.0164)
Similarity-Based (Power Decay)	**0.8891 (0.0609)**	0.5126 (0.0169)	0.2572 (0.0127)
Similarity-Based (Exponential Decay)	0.8988 (0.0610)	0.5109 (0.0169)	0.2564 (0.0127)
Similarity-Based (K Nearest Neighbors)	0.9300 (0.0629)	**0.5091 (0.0169)**	**0.2530 (0.0126)**

Table 5: The mean distances of the distributions estimated by the various smoothing methods from the corresponding Gold Standard distributions in Simulation 1 (standard errors are shown in parentheses; the best-performing method under each distance metric is shown in bold face).

outperformed by the Uniform method – which uses none of the information contained in the training corpus – after the first section of training data. In other words, it was not until at least 5 million words had been encountered that the probability estimates arrived at through maximum likelihood were more reliable than simply assuming that every word was equally likely to occur at each point. Figure 11 shows the performance of the Simple Backoff, Witten-Bell and Jelinek-Mercer methods, all of which attained approximately the same level of performance, which represents a slight improvement on the previously mentioned methods. Finally, Figure 12 shows the performance of the Absolute Discounting, Kneser-Ney and Modified Kneser-Ney techniques, along with the similarity-based methods. As can be seen, the similarity-based methods reliably get closer to the Gold Standard probability estimates and, as confirmed by our earlier statistical analysis, this difference was significant for the Power and Exponential decay implementations.

In order to get a sense for how good the fixed parameter estimates were for the models for which they were specified, we calculated the percentage gain in performance, as measured by the Kullback-Leibler metric, for using the optimized free parameter estimates over the fixed one. The advantage for using the free over the fixed parameter values was 7.96% for Absolute Discounting, 5.11% for Kneser-Ney, and 6.12% for Modified Kneser-Ney. While using an optimization process clearly yields some performance gain, these figures tend to suggest that the fixed values of the parameters that have been specified are doing a reasonable job of estimating appropriate values.

Decay Rate

How did the decay weights over the nearest neighbors affect the performance of the similarity-based smoothing methods? In order to address this question, we focused on the power-decay implementation, as this was the best-performing technique under the Kullback-Leibler distance metric. At each of the 6 points through the training data we systematically explored 100 values for the decay parameter, p_1, and for each value used simulated annealing to find the optimal values for the remaining free parameters, p_2-p_4. Clearly we would expect the smoothing method to get closer to the Gold Standard distributions on average, with increasing training data. Therefore, in order to normalize the scores in order to control for this factor, we recorded the z-score of the distances for each value of the decay parameter, within the population of scores at that point in the training data. The resulting plot is shown in Figure 13. As can be seen, an interesting pattern emerges. Apart from the first point in the training data, all the other profiles have a well-defined minimum value as a function of the decay parameter, suggesting that there is an optimal degree of generalization across the nearest-neighboring distributions which balances a lack of information (undergeneralization) against the noise introduced by weighing remote distributions too heavily (overgeneralization). In addition, there is a suggestion that a broader generalization gradient is more optimal as more training data becomes available: the minimum of each curve seems to increase as a function of the amount of training data. In summary, a generalization gradient in which the nearest-neighbor to a distribution receives a weight of approximately 0.5 under the power-based decay seems to be generally optimal.

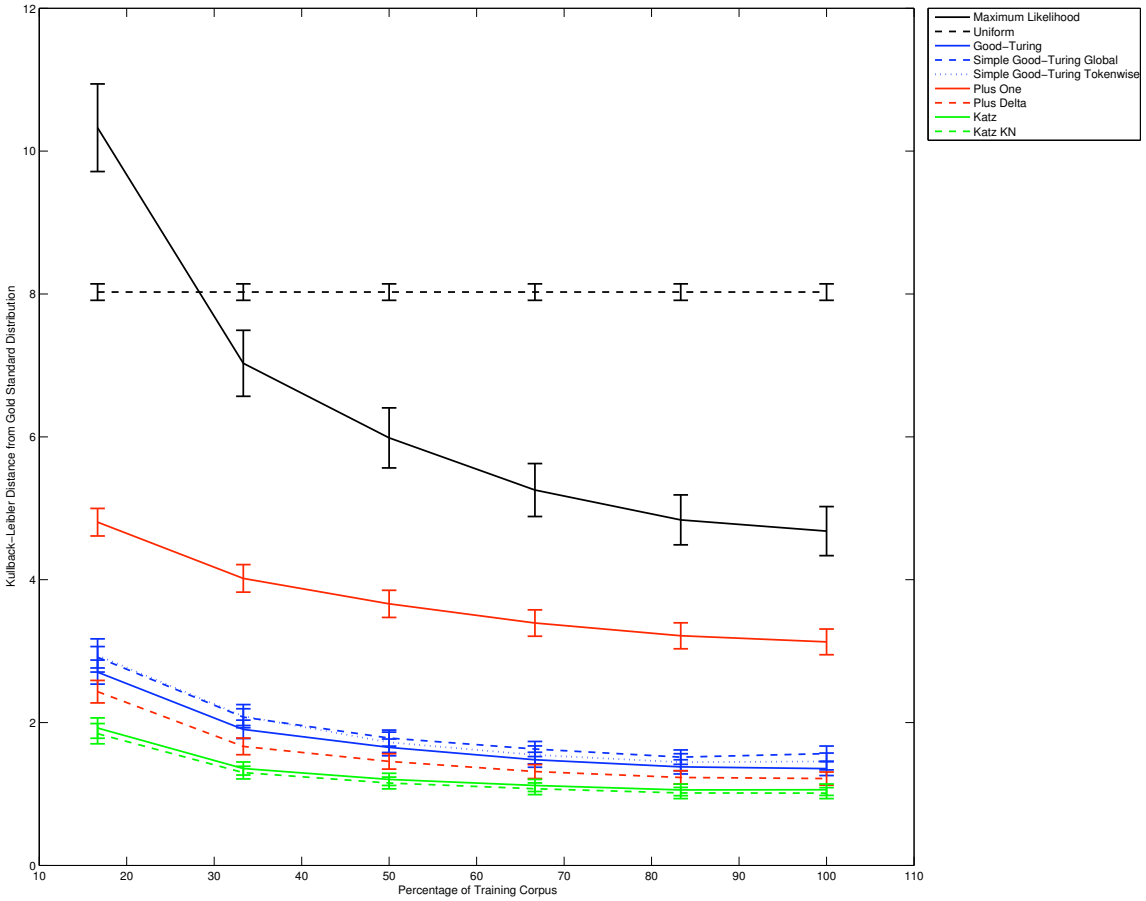

Figure 10. Performance of the Maximum Likelihood, Uniform, Good-Turing, Additive and Katz methods under the Kullback-Leibler metric in Simulation 1.

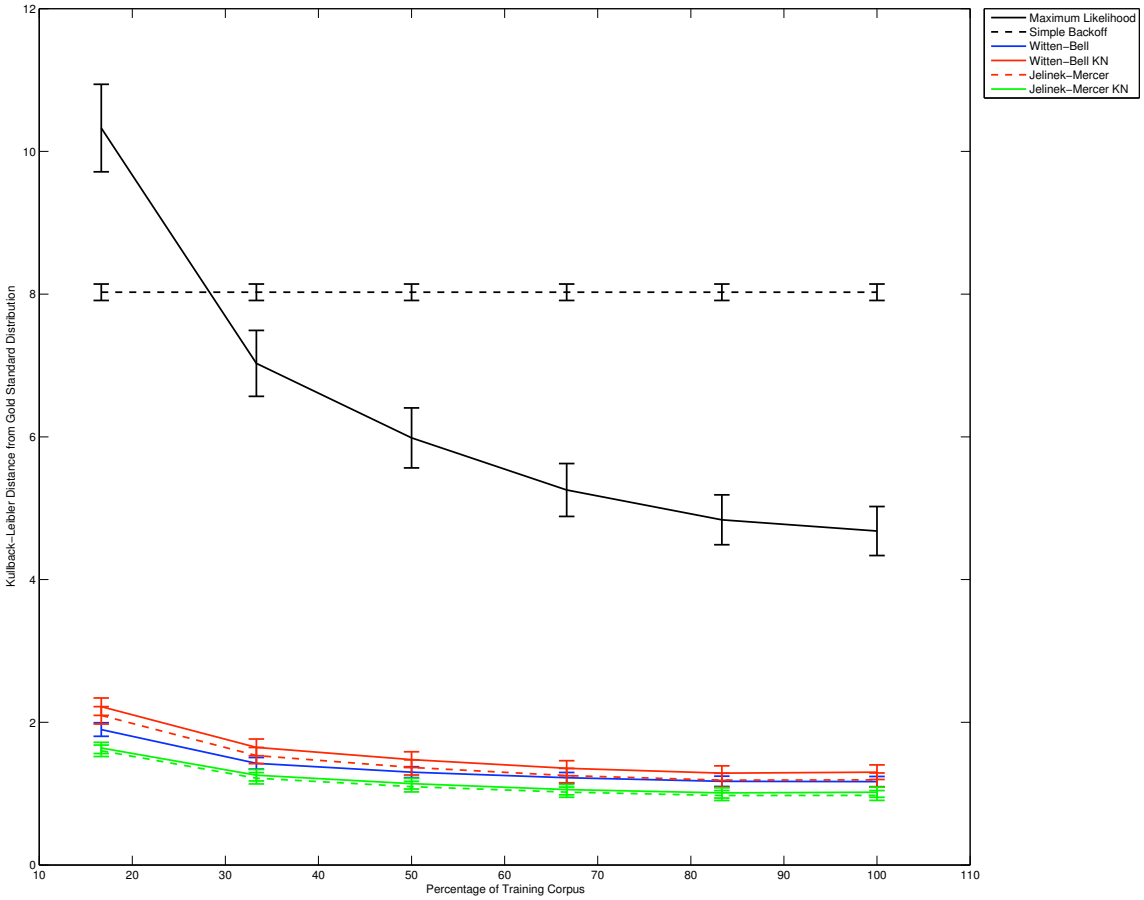

Figure 11. Performance of the Maximum Likelihood, Simple Backoff, Witten-Bell and Jelinek-Mercer methods under the Kullback-Leibler metric in Simulation 1.

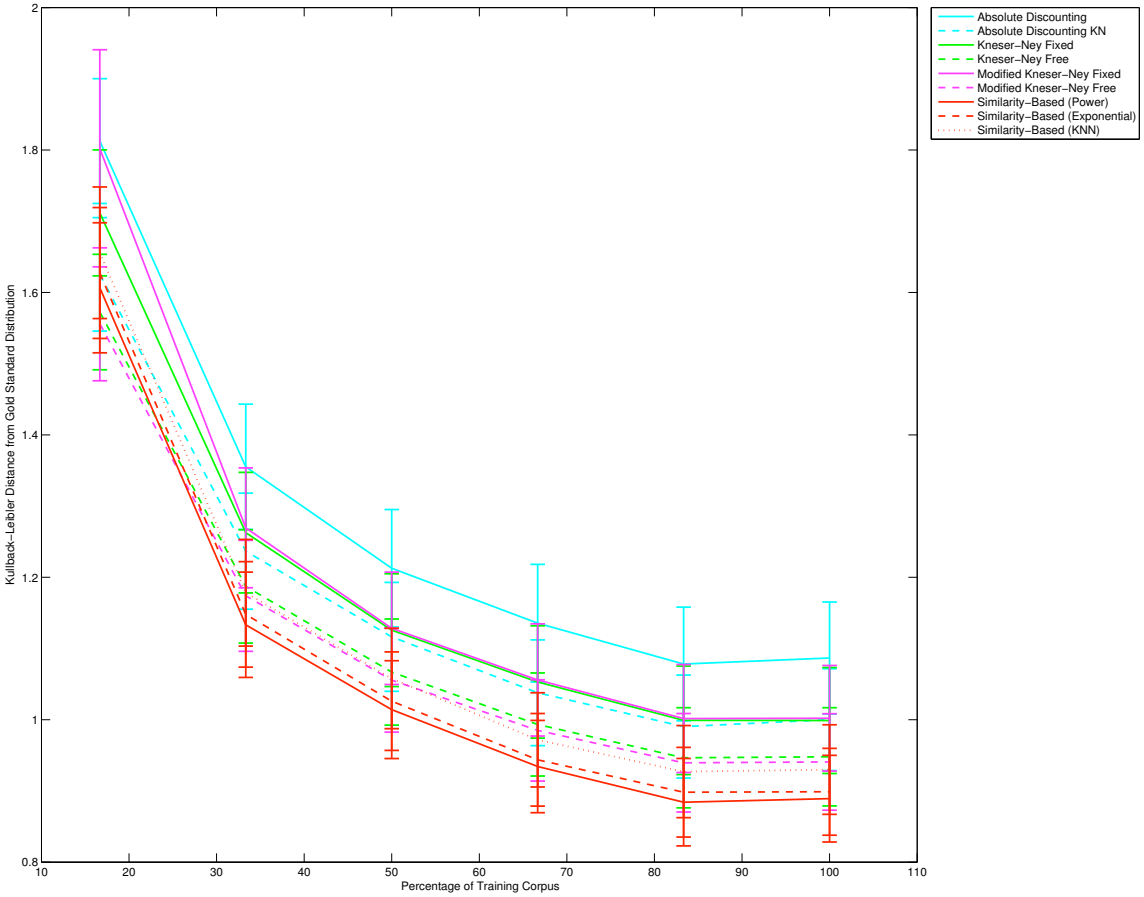

Figure 12. Performance of the Absolute Discounting, Kneser-Ney, Modified Kneser-ney and Similarity-Based methods under the Kullback-Leibler metric in Simulation 1.

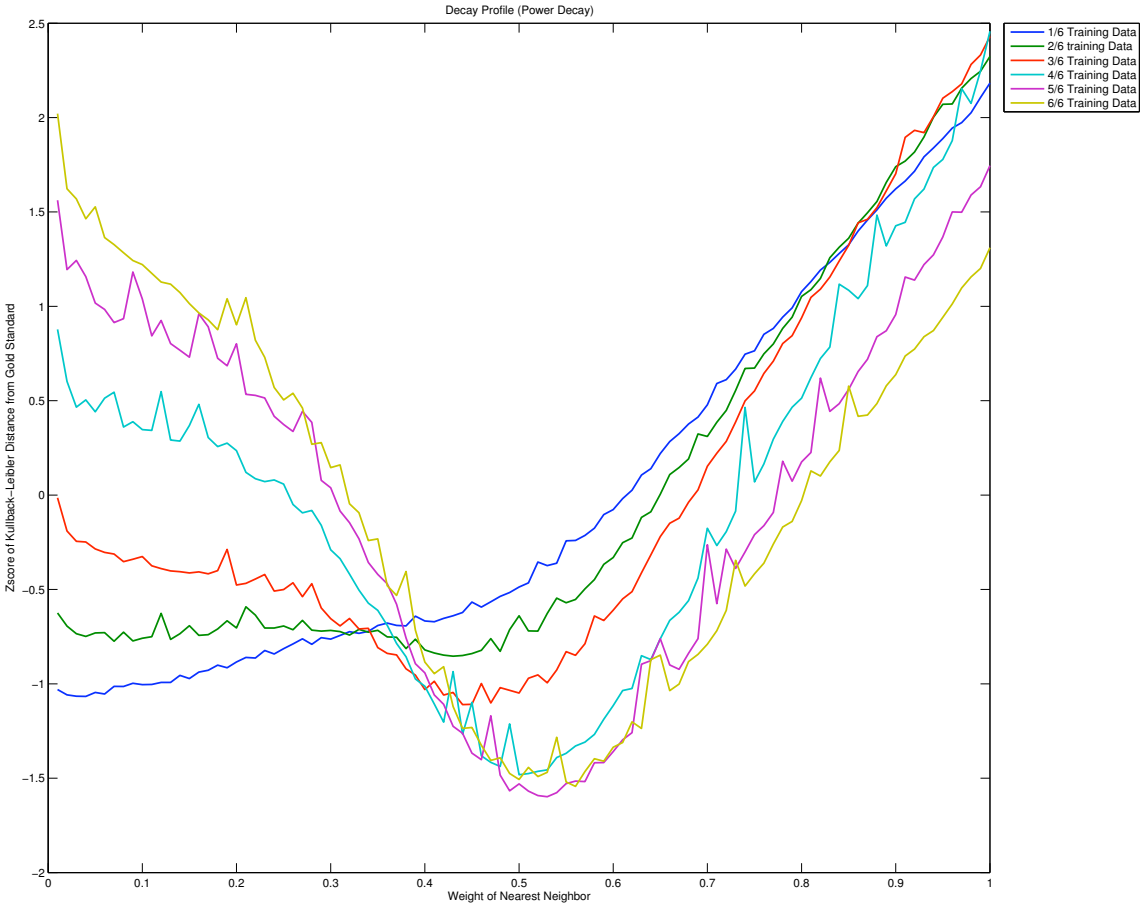

Figure 13. Plot of the z-score of the distance from the Gold Standard distributions reached, on average, under the Kullback-Leibler distance metric, as a function of the amount of weight that the nearest-neighbor of a sampled word receives (this is determined by the value of p_1). The optimal level of generalization seems to occur when the nearest neighbor of a target word receives an activation of 0.40-0.50. In addition, it seems to be adaptive to generalize slightly more liberally as more training data becomes available: this could be due to (i) the increasing reliability of the distributional metric, and (ii) the increasing reliability of the information contained in the neighboring distributions (i.e. they will be less affected by data-sparsity).

Summary and Discussion

We have shown that when it comes to estimating transitional probability distributions that the similarity-based models we have proposed are able to do so more accurately than other smoothing methods. This result suggests that similarity-based generalization does indeed offer a powerful mechanism by which problems of data-sparsity in language acquisition can be reduced, and is consistent with the initial success of this kind of approach reported by Dagan, Lee and Pereira (1999). This finding is also consistent with our general thesis that similarity-based information could make a valuable contribution to human language learners attempting to overcome the fundamental problem of data-sparsity.

The success of the similarity-based methods is particularly striking given the effort taken to compile and implement a comprehensive list of current smoothing methods. The success of similarity-based methods in the present context should not, however, be taken as evidence that the information contained in the unigram (or the Kneser-Ney) distributions is unimportant, or even irrelevant. The success of the similarity-based techniques is dependent to some degree on the information contained in the unigram distribution, which was used to cover the neighboring distribution in Simulation 1 (see Equation 38). The parameter controlling the weight of the covering distributions was estimated to be reliably greater than 0 in all cases, suggesting that an optimal model is one that uses information derived from *both* similarity-based information *and* the marginal probability of observing particular word types. It is an interesting question, given the relative success of the Kneser-Ney models, which were the second-best group of models according to the KL metric, to ask whether the performance of the similarity-based models could be improved by using the Kneser-Ney distribution as the covering distribution for the similarity-based models. We turn to this question later.

A number of additional questions also remain to be answered after Simulation 1. In this simulation we proceeded by using a simulated annealing algorithm, which relied on access to some of the Gold Standard data in order to set the free parameters of those methods that used them. We did this deliberately, as we wished to assess the absolute quality of the various smoothing methods without the possible confound of suboptimally estimated parameter values. But this is not at all like the situation faced by a human language learner who, of course, has no access to the 'real' probabilities of events in language. We therefore wished to consider a situation in which smoothed distributions had to be generated without access to any of the Gold Standard information. Many of the smoothing methods we considered have ways of doing exactly this, for example by using information about the frequency of a conditioning event, or by examining the frequency of frequency distributions of a corpus (these are the so-called 'fixed' versions of the smoothing techniques we have discussed). With reference to this question, one may be concerned that the optimal parameter values for the similarity-based methods are not subject to reliable regularities, so that attempting to set them through heuristic methods using only the data previously encountered by a language learner might be much harder than for the other techniques. Clearly, what is required, then, is some way of estimating the parameters for the similarity-based models without access to any 'illegitimate' data such as that contained in the Gold Standard distributions. In order to address

these concerns, we conducted Simulation 2.

Simulation 2

Simulation 2 was designed to more closely reflect the situation faced by a human language learner attempting to estimate the probability of words based only on information contained in the language they have previously encountered. In this case, information contained in the Gold Standard distributions – which was used in Simulation 1 in order to help determine the values of the free parameters of the smoothing techniques that had them – should clearly not be accessible to the smoothing models. A number of other changes were also made to the design of Simulation 1 in order to further explore the conditions under which similarity-based smoothing can be expected to achieve a high level of performance. In Simulation 2 the training data was defined to be the *first* 30% of the BNC – as opposed to the *last* 30% as in Simulation 1 – in order to minimize the possibility that the results of Simulation 1 were due an idiosyncrasy of the corpus. In addition, a larger set of distinct sample words – 500 instead of 200 – which included much lower-frequency items than in Simulation 1 was used in order to improve the statistical power of our analysis (this was possible because of the computational savings derived from not using the simulated annealing algorithm in order to optimize parameter settings), to check that the results of Simulation 1 were not due to some accidental property of the originally sampled words, and to explore the behavior of the smoothing models over a wider range of boundary conditions.

Most of the pre-existing smoothing models specify ways in which their parameter values can be set if an optimization process is not used. However, no equivalent method for similarity-based smoothing has been established yet. Therefore, we began by exploring the 'optimal' parameter values that were found for the similarity-based model in Simulation 1 through the simulated annealing search process, in order to determine whether these values exhibited any regularities that could be used to set appropriate values based only on the information contained in previously encountered input (i.e. based only on information contained in the training corpus). Based on this analysis we derive a method for setting the parameters of the similarity-based model, and then test the effectiveness of this method against the other smoothing methods.

Analysis of Parameter Values from Simulation 1

In order to determine whether there were any regularities in the free parameter values that were estimated for the similarity-based method in Simulation 1, and hence whether there might be a simple way of setting these parameters without having access to the Gold Standard distributions, we examined the ways in which each of the 4 parameter values changed with increasing training data. We restricted our analysis to the k nearest neighbor implementation of the similarity-based approach, even though it wasn't the best-performing similarity-based model according to the KL metric. We did this for several reasons. One advantage of the k nearest neighbor approach is that it makes the p_1 parameter very interpretable, as it simply specifies the number of nearest neighbor distributions that are to be summed over; in the case of the power- and exponential-decay models, p_1 is not as immediately interpretable. In addition, use of the k nearest neighbor model avoided the

need to restrict how deep the process of summation needed to go (for example, in Simulation 1 we had to restrict it to 1000, and we were concerned that setting a limit to the summation process could have unduly impaired the performance of the model), and also meant that we ended up with a more computationally efficient model. The use of the k nearest neighbor implementation does, however, mean that we are likely to obtain a conservative estimate of the performance of similarity-based approaches in general.

The mean value assigned to each parameter as a result of the simulated annealing optimization process in Simulation 1, plotted against the amount of training data used, is shown in Figure 14. From some preliminary exploration we thought it likely that parameters 1-3 would have a less significant impact on the level of smoothing performance than parameter 4. In addition, the values of parameter 1-3 appeared to vary less as a function of the amount of training data than parameter 4. We therefore decided to try using representative but constant parameter values for the first three parameters, but to tailor the value of parameter 4 based on information derived from the training data. Note that in doing this our goal was not simply to maximize the performance of the similarity-based model (if this had been the case we would have tried varying the values assigned to parameters 1-3 as a function of the training data), but rather to demonstrate that a relatively simple method of setting the parameter values could achieve competitive levels of performance relative to other smoothing techniques.

In accordance with this approach we fixed parameter 1, which controls the number of nearest neighbor distributions that are summed across, to 25; we fixed parameter 2, which controls the relative weight of the covering distribution relative to the neighboring distribution, to 0.75; and parameter 3, which controls the rate at which empirically observed frequencies are discounted, to 1.2 (see Table 2 for descriptions of the exact function of each of the parameters). Then we explored how parameter 4 should be set. Prior research has found that the frequency of the history which defines a distribution has a large impact on the degree to which smoothing should be applied (for example, see Chen and Goodman, 1996, 1998). We therefore decided to examine whether the same kind of regularity applied to the similarity-based smoothing data we had collected in Simulation 1. If a reliable regularity could be observed between some measure of a distribution (e.g. its frequency, or how 'populated' the distribution is), then this could be used as a way to assign an appropriate parameter value to parameter 4 in cases in which the Gold Standard distributions were unavailable. We explored a number of possibilities, but in general the best predictor of the smoothing parameter p_4 was the type:token ratio of the directly observed bigram distribution (this is equal to the reciprocal of the mean non-zero value in the distribution). Because the number of distinct types observed in a distribution is less than or equal to the number of tokens in that distribution, this quantity is constrained to lie between 0 and 1 (inclusive). In addition, this quantity has a pleasing theoretical interpretation as a probability: it can be regarded as an estimate of the probability that a previously unseen word type will be observed to occur in the distribution (Witten and Bell, 1991). Clearly when data-sparsity is a serious problem the probability of a new word type occurring in the distribution will be high and hence smoothing should be fairly aggressive. However, once the probability of observing a new word type is low, then this offers evidence that the distribution has become relatively mature, and hence that less smoothing will be required. Figure

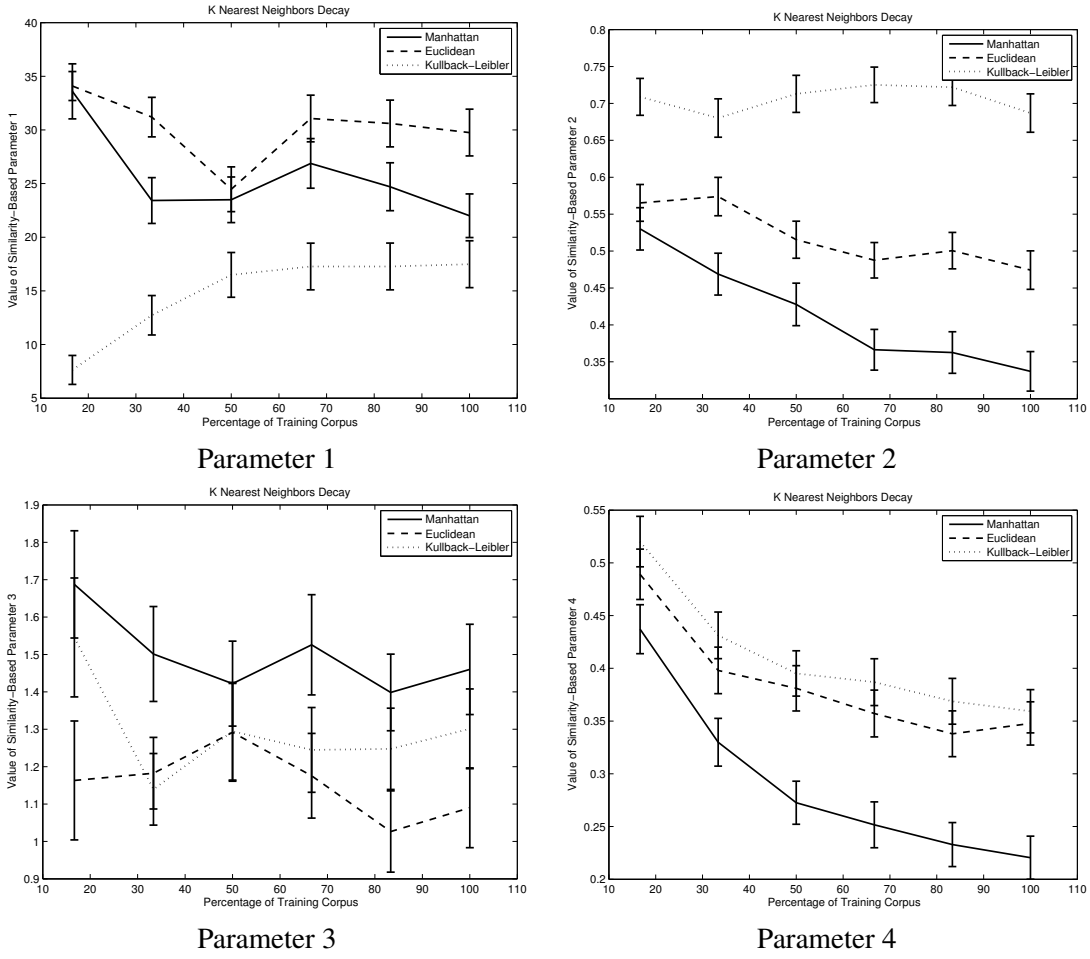

Figure 14. Plots showing how the mean value of each of the 4 similarity-based parameters changes with increasing training data. Each plot shows the mean value of one of the parameters as a function of the percentage of training data used in Simulation 1, under each of the 3 metrics used to measure distance from the Gold Standard distributions.

15 shows a plot of the 'optimal' value for parameter 4 as determined by the simulated annealing process in Simulation 1, against the type:token ratio for the corresponding empirically observed bigram frequency distribution. As can be seen, there was a reliable correlation between these two quantities ($r = 0.38$). We therefore decided to use the type:token ratio of directly observed bigram frequency distributions in order to determine the value that should be assigned to p_4 in Simulation 2. Rather than use a regression model, we simply set the value of p_4 to be the value of the type:token ratio for simplicity (again, this could have led to less than optimal performance for the model, but our primary concern was to demonstrate the simplicity of finding acceptable parameter estimates for the similarity-based approach, rather than to produce the best possible similarity-based model).

Implementation of Models

Because the optimization process used to set the parameter values for the *free* models in Simulation 1 relied on access to information contained in the Gold Standard distributions, and this information was explicitly denied the models in Simulation 2, all the models used in Simulation 2 were *fixed* versions. Where a *fixed* version of a model had not been specified in the literature, this model was excluded from Simulation 2.

Results

Simulation 2 was similar to Simulation 1, except that rather than using simulated annealing and access to the Gold Standard distributions in order to arrive at 'optimal' free parameter values for the smoothing methods, we denied the smoothing methods access to the Gold Standard distributions in order to more accurately model the situation faced by a human language learner. This meant that the smoothing models had to rely on heuristic methods for setting the values of any parameters they utilized. The parameter values of the similarity-based model were set as described in the previous section; the heuristic methods employed by the other models are described earlier in this article. We also made two other changes to the basic design of Simulation 1. First, the training set was defined to be the first 30% of the BNC, instead of the last 30% as it was in Simulation 1. This was to check that the relative performance of the smoothing methods was not dependent on some potentially arbitrary aspect of the training and/or test corpus. Second, it became possible to use a larger and distinct set of sample words because of the considerable computational savings involved in not using simulated annealing to estimate the values of the free parameters. Therefore, we used 500 instead of 200 sample words to define the distributions to be estimated. These sample words exhibited a greater range of frequencies, which allowed us to examine whether the similarity-based smoothing method still performed well when a distribution had only been observed a few times. In Simulation 1 the lowest frequency sampled word occurred 210 times in the BNC; in Simulation 2 the lowest frequency sampled word occurred only 10 times in the BNC (this meant that 6 of the 500 words never occurred in the training corpus at all, indicating that the smoothing methods were being tested over the full range of possible frequencies, from the most frequently occurring items right down to items that never occurred at all). In addition, in order to explore the potential advantage of the Kneser-Ney distribution over the standard unigram distribution, we implemented two

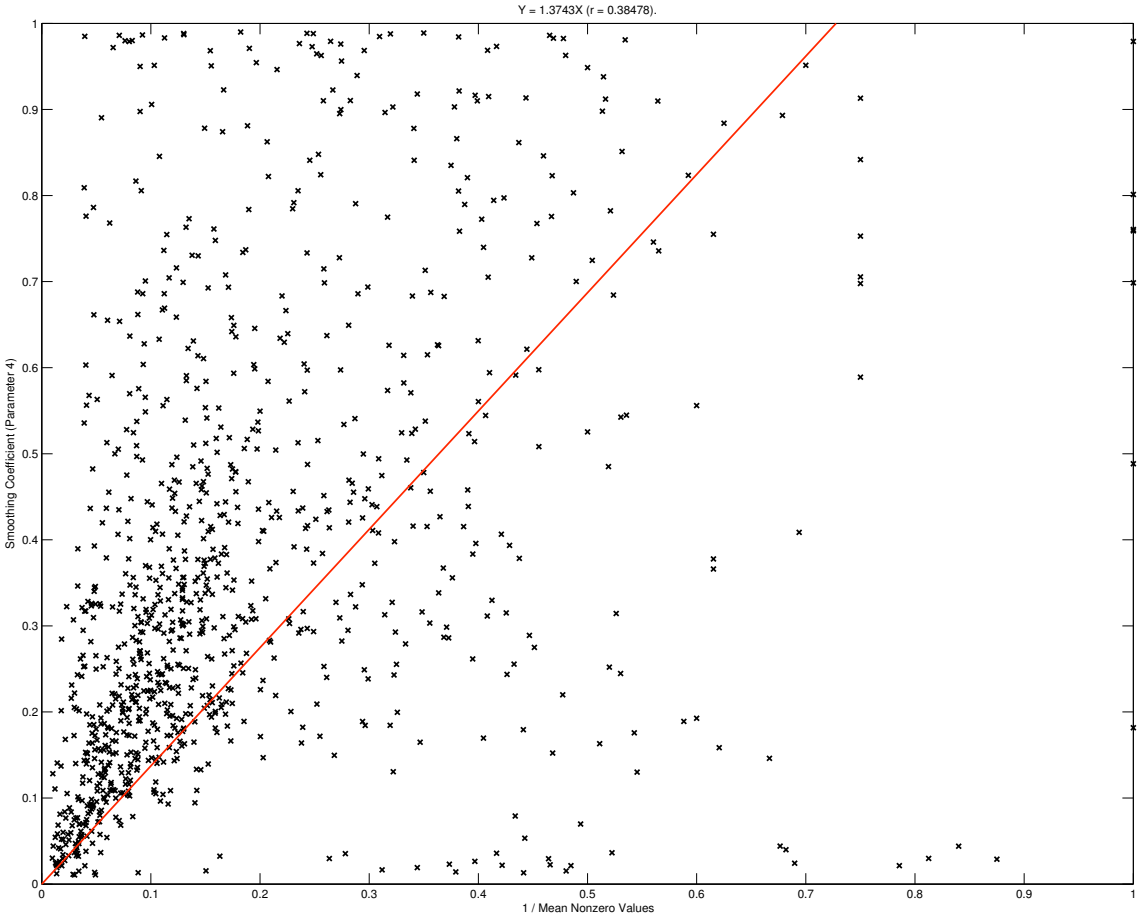

Figure 15. Scatterplot of the 'optimal' parameter values for the smoothing coefficient (p_4) derived through simulated annealing in Simulation 1, plotted as a function of the type:token ratio for the corresponding distributions. The strong correlation suggests that this statistic could be used to set appropriate values for this parameter when only information contained in previously observed linguistic input (i.e. the training corpus) is available.

versions of the Similarity-Based approach, one which used the unigram distribution as its covering distribution, and one which used the Kneser-Ney distribution as its covering distribution (both versions used k nearest neighbors decay, as discussed).

Figure 16 shows the performance of each smoothing method in Simulation 2 as a function of the amount of training data used (the mean KL distance from the Gold Standard distributions is plotted). As can be seen, the first and second methods in terms of performance at the end of training were the two similarity-based implementations. The next best method at this point was the Modified Kneser-Ney method, which is consistent with the results of Chen and Goodman (1998), in which the Modified Kneser-Ney method outperformed all the smoothing methods they investigated. The similarity-based model covered by the unigram distribution did not perform significantly better than the Modified Kneser-Ney method ($t_{499} = 1.45$, $p = 0.15$). However, the similarity-based method covered by the Kneser-Ney method did clearly outperform the MKN model ($t_{499} = 5.66$, $p = 2.50 \times 10^{-8}$). It is particularly striking that the similarity-based methods performed so well given that we set the parameters in such a simple fashion, without any attempt to vary the values of p_1-p_3. It is quite possible that the absolute performance of a similarity-based model could be increased considerably with a more careful attempt to determine suitable parameter values based on the input data.

Smoothing Method	Kullback-Leibler	Manhattan	Euclidean
Maximum Likelihood	9.5586 (0.4374)	**0.7341 (0.0197)**	0.4351 (0.0162)
Katz	2.0164 (0.0930)	0.7637 (0.0203)	0.4286 (0.0161)
Katz (Kneser-Ney)	2.0646 (0.0970)	0.7824 (0.0212)	0.4312 (0.0164)
Simple Back-Off	2.0909 (0.0854)	0.9865 (0.0155)	0.4518 (0.0158)
Witten-Bell	2.5725 (0.1209)	0.7489 (0.0200)	0.4360 (0.0163)
Absolute Discounting Fixed	2.0122 (0.0908)	0.7476 (0.0202)	0.4233 (0.0158)
Kneser-Ney Fixed	2.0201 (0.0943)	0.7857 (0.0215)	0.4290 (0.0164)
Modified Kneser-Ney Fixed	1.9894 (0.0921)	0.8132 (0.0218)	0.4289 (0.0164)
Similarity-Based (Unigram Covering)	1.9594 (0.0914)	0.7361 (0.0195)	**0.4171 (0.0154)**
Similarity-Based (Kneser-Ney Covering)	**1.8944 (0.0883)**	0.7454 (0.0197)	0.4179 (0.0156)

Table 6: The mean distances of the distributions estimated by the various smoothing methods from the corresponding Gold Standard distributions in Simulation 2 (standard errors are shown in parentheses; the best-performing method under each distance metric is shown in bold face).

Influence of Covering Distributions

One of the goals of Simulation 2 was to examine whether the unigram or Kneser-Ney distribution constituted a more reliable covering distribution when used in similarity-based smoothing. To this end we implemented two versions of our similarity-based model, one which relied on each

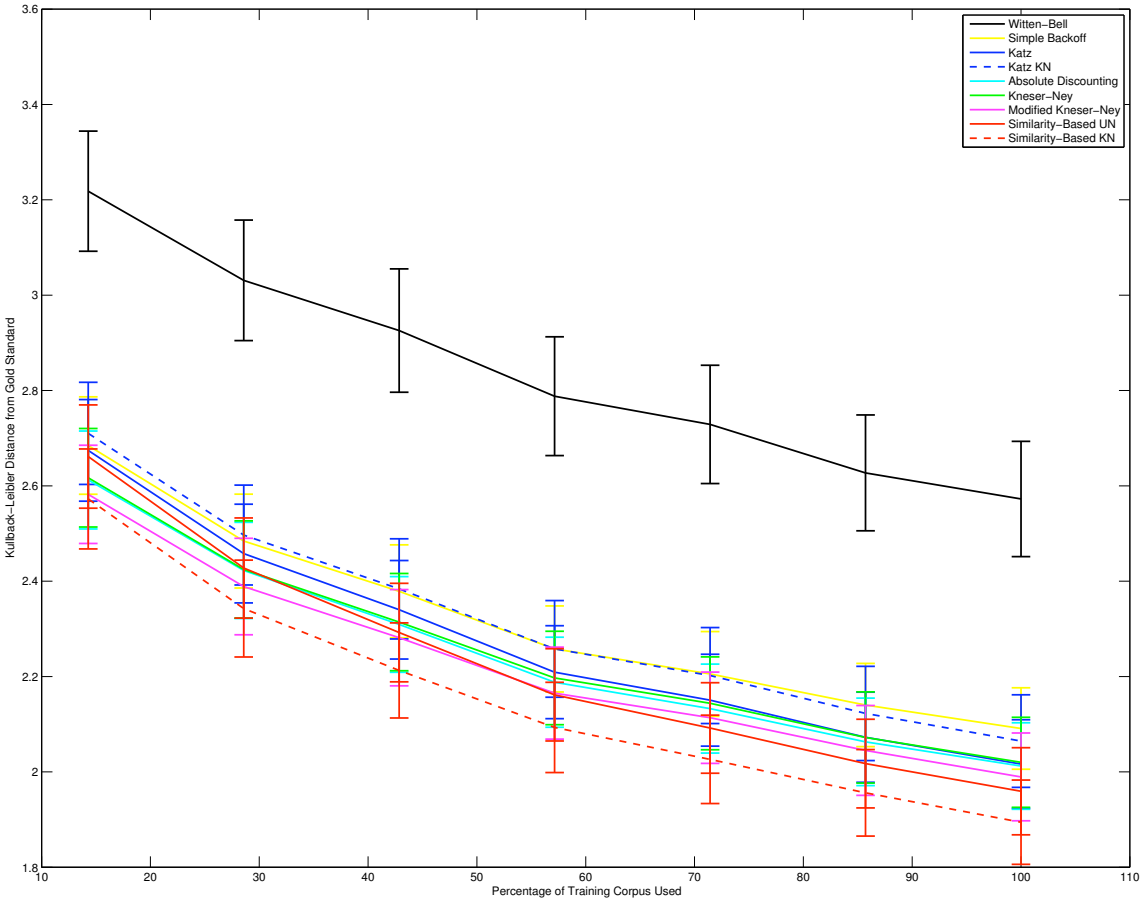

Figure 16. The mean distance of the smoothing methods from the Gold Standard distributions in Simulation 2. Distances are measured according to the Kullback-Leibler metric. The two similarity-based methods were the best-performing methods.

of these two possible covering distributions. The similarity-based model which used the Kneser-Ney distribution performed significantly better than the version using the unigram distribution as its covering distribution ($t_{499} = 9.42$, $p = 1.66 \times 10^{-19}$). This suggests that the Kneser-Ney distribution does indeed provide a more reliable source of information about the probability of observing a word in a given context, as argued by Kneser and Ney (1995). However, this conclusion is complicated by two additional observations. First, that there was no reliable difference in performance between the Absolute Discounting and Kneser-Ney models ($t_{499} = 0.50$, $p = 0.62$) – which are identical models except that they use the unigram and the Kneser-Ney distributions, respectively. And second that the version of Katz smoothing which used the Kneser-Ney distribution as its back-off distribution performed worse than the standard version of the model, which uses the unigram distribution ($t_{499} = 3.15$, $p = 0.0017$). Clearly, claims about the relative merits of a distribution need to be made relative to the specific way in which that distribution is used by a particular smoothing algorithm.

It is hard to make any absolute claims about the importance of a covering distribution, and particularly so in the case of similarity-based smoothing because the exact importance is likely to depend at least on (i) how much training data has been encountered (more training data makes it less likely that important co-occurrences will fail to be observed in the nearest neighboring distributions), and (ii) the number of neighboring distributions that are summed across and the manner in which this is done. However, in order to get some information on this issue, we explored the performance of each of the two versions of the similarity-based smoothing model using a range of different covering distribution parameters (p_2). The range of covering parameter values explored were as follows: 0.00, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00, 1.11, 1.25, 1.42, 1.67, 2.00, 2.50, 3.33, 5.00, 10.00. These values cover the entire spectrum of importance assigned to the covering distribution, from the case in which a covering distribution is not used at all, to the situation in which the covering distribution receives 10 times more weight than the neighboring distribution derived through similarity-based generalization. Based on this data, we examined the mean distance each of the two similarity-based implementations reached from the Gold Standard distributions as a function of the value of the covering parameter. This data is shown in Figure 17. As can be seen, the similarity-based model relying on the Kneser-Ney covering distribution was closer to the Gold Standard distributions on average, for all values of the covering parameter p_2. This again confirms the idea that, in general, the Kneser-Ney distribution contains valuable information about the probability of encountering a word in a given context. A second point to observe from this analysis is that the model's performance is maximized (i.e. mean distance from the Gold Standard distributions is minimized) when the neighboring distribution and the covering distribution receive approximately equal weight ($p_2 \approx 1$). This supports the idea that the two types of distributions – the one derived from similarity-based generalization and the other derived from marginal counts of either words tokens or types – provide distinct sources of information that can be combined in order to complement one another.

The final analysis we performed was to examine how the difference in performance between the similarity-based Kneser-Ney model and the Modified Kneser-Ney model varied as a function of the rank frequency of the sample word defining each distribution (this is a proxy for their absolute

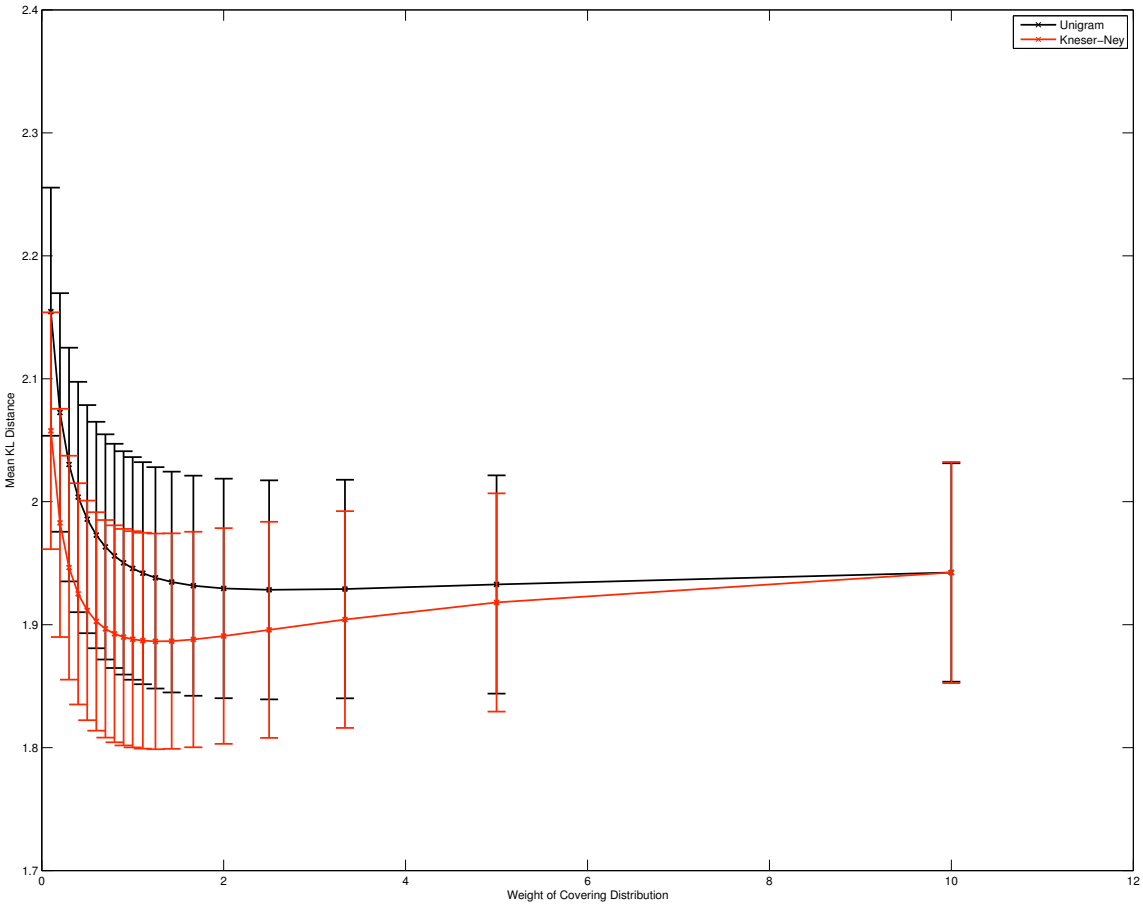

Figure 17. The mean distance from the 500 Gold Standard distributions as a function of the covering parameter (p_2), for both the unigram and the Kneser-Ney covering distributions. Optimal performance appears to be achieved when the neighboring distribution and the Kneser-Ney distribution are combined with approximately equal weight.

frequency and is highly correlated with log frequency). This data is shown plotted in Figure 18. As can be seen, there is relatively little difference in performance between the two models for the most frequent 250 words or so, but after this the similarity-based model starts to clearly dominate. This pattern was perhaps to be expected: when a distribution has been observed many times – i.e. is highly frequent – then any smoothing algorithm will base its probability estimates primarily on the information contained in the raw bigram frequency distribution. In this case, most smoothing methods will be equally good for high frequency items. It is only when a distribution becomes relatively infrequent that the smoothing coefficients of most models will be non-negligible, and so it is only at this point that differences in the quality of smoothing algorithms can become apparent.

Summary and Discussion

In Simulation 2 we established a simple method for setting the parameters of the similarity-based models, and showed that using the parameters derived from this method allowed the similarity-based models to produce more accurate estimates of transitional probabilities in language than other existing models that have been proposed. This was particularly impressive given that 3 of the similarity-based model's parameters were simply fixed to constant values and no attempt was made to set the values for these parameters based on the smoothing context. As well as showing that similarity-based information can be exploited based only on information contained in previously observed linguistic input, this also shows that the excellent performance of the similarity-based approach in Simulation 1 was not a result of over-fitting or highly optimized parameter estimates. In short, the performance of the similarity-based framework we have described appears to be relatively robust, which adds further support to our claim that this information could be readily exploited by human language learners.

Simulation 3

Performance in a language modeling task is the standard method used to evaluate smoothing techniques (Charniak, 1993; Chen and Goodman, 1996, 1998; Manning and Schütze, 1999). So far in this article we have used an alternative methodology in which we have attempted to estimate entire conditional probability distributions rather than individual probabilities. We elected to use this method because it allows for extensive parameter optimization to take place using only a limited number of test distributions; this allowed us to analyze the impact of the parameter values of the various smoothing models on smoothing performance in Simulations 1 and 2. However, in order to get a validation of these results we decided in Simulation 3 to evaluate the smoothing methods which have exhibited the highest levels of performance in a more standard language modeling task. This also allowed us to address a potential limitation of the Gold Standard methodology we had hitherto adopted, which we now discuss.

Although the sample of words we used in Simulations 1 and 2 were deliberately selected to cover a wide range of frequencies, these samples do not accurately reflect the frequency distribution of language in general. Zipf's Law implies that most tokens encountered in a real sample of language will tend to have high frequencies, whereas in our samples each frequency range for the

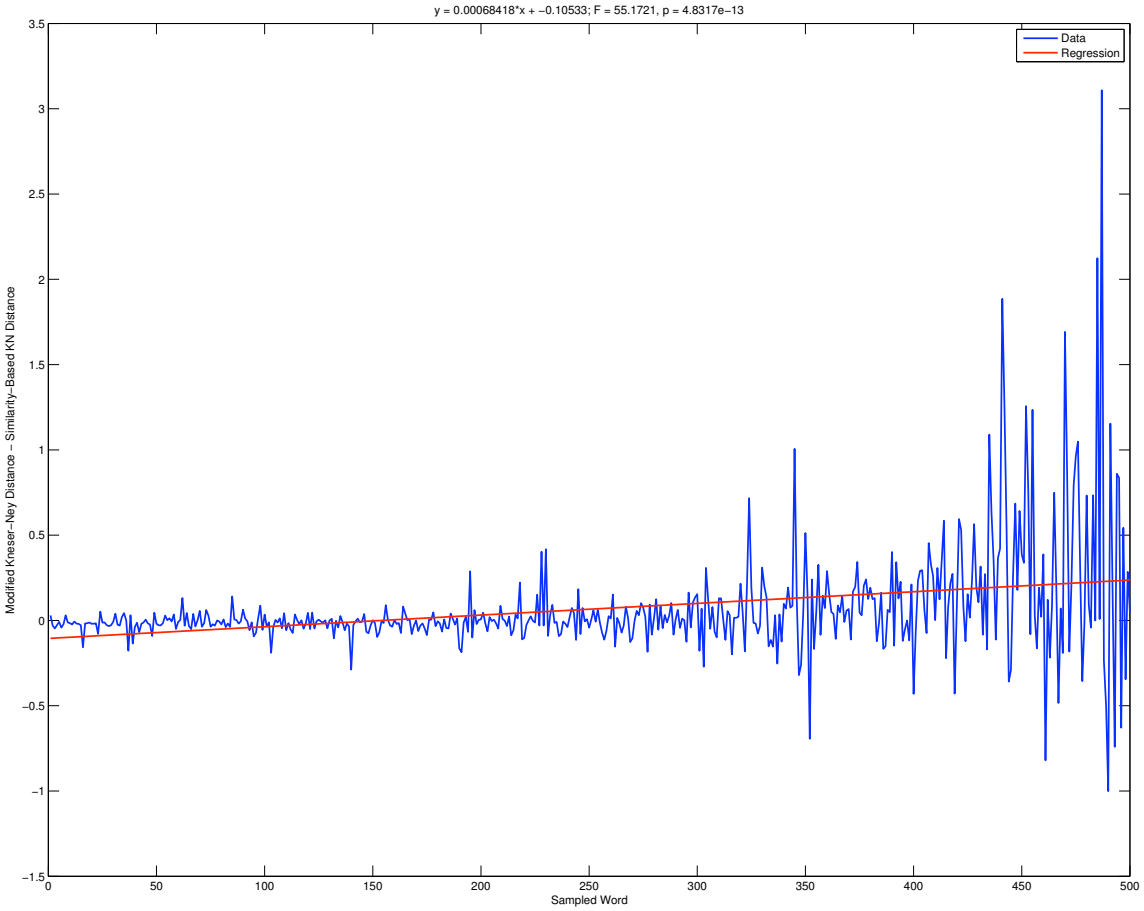

Figure 18. Plot of the mean difference between the Kullback-Leibler distance from the Gold Standard Distributions for the Modified Kneser-Ney and Similarity-Based KN methods (positive values indicate the Similarity-Based method was outperforming Modified Kneser-Ney smoothing). There was a significant trend for the difference in performance to become greater as the frequency of the sampled word defining the distribution became less frequent ($p = 4.83 \times 10^{-13}$). This is likely to be because, in general, smoothing algorithms tend to be more aggressive with lower frequency distributions, and hence it is here that the differences in quality between smoothing methods can become apparent.

words received equal representation. Although this was a deliberate design choice intended to test the performance of the various smoothing methods over a wide range of frequencies, it does mean there is the possibility of a bias in the results of Simulations 1 and 2. It could be, for example, that there was a systematic bias in the performance of the similarity-based technique relative to the other smoothing methods, such that it smoothes the distributions associated with low frequency words very well, but performs relatively poorly when it comes to smoothing the distributions associated with high frequency words. This means that relative performance of the smoothing techniques could conceivably change when it came to a 'real world' language modeling task. Language modeling as a paradigm avoids this objection because the probabilities that it requires to be estimated are drawn directly from real samples of language use, which will of course reflect Zipf's Law.

Models

We decided to evaluate the performance of the similarity-based model covered by the unigram and the Kneser-Ney distributions (using the same parameter values used in Simulation 2), and also the Absolute Discounting, Kneser-Ney and Modified Kneser-Ney methods. These last three techniques were selected because they were, across the first 2 simulations, the best-performing methods after the similarity-based models.

Language Sample

In Simulation 3 we used the entire BNC corpus to provide training data for the language models arising as a result of the smoothing methods we decided to investigate. We therefore needed to use a separate sample of language with which to test the language models. We decided to use text drawn from the BBC News website to act as our test sample because it covers a wide range of topics including politics, sport and entertainment, is written in British English (like the BNC), and is readily available in abundant quantities. A script was written to interrogate the front page of BBC News website (`http://news.bbc.co.uk/`), and to extract URLs from the articles linked to from this page. The URLs of approximately 1,000 news articles were identified in this fashion. These URLs were randomized in order and the text from each article was extracted in order to generate a reasonably-sized corpus of text.[10] We used the first 50,000 tokens of this sample as the test corpus for Simulation 3. Previous studies of language modeling suggest that this is a sufficient sample size in order to allow for accurate discrimination between the performance of different smoothing methods.

Not all of the bigram transitions in this sample were relevant to our concerns. Some of the transitions occurred between word types that were not in the lexicon generated from the BNC (inspection revealed the majority of these types were proper nouns, such as names). Because these transitions reflected limitations in the lexicon used and not limitations in the smoothing models, these transitions were excluded from the analysis. A total of 4.11% of the tokens in the language sample did not occur in the BNC's lexicon and so were excluded accordingly. In addition, some of the transitions were to words which were not in the first 19,999 most frequent word types in

[10]The news text was downloaded on December 27th 2006.

Method	Mean Log Probability	Mean Perplexity
Unsmoothed	10.6324 (0.0509)	1587.34
Absolute Discounting	8.4235 (0.0235)	343.34
Kneser-Ney	8.3649 (0.0225)	329.67
Modified Kneser-Ney	8.3482 (0.0222)	325.88
Similarity-Based Unigram	8.3901 (0.0231)	335.48
Similarity-Based Kneser-Ney	8.3428 (0.0224)	324.66

Table 7: Language modeling results from Simulation 3.

the BNC. Because our language models used 20,000 element vectors to represent the estimated conditional probability distributions, and the 20,000th element is the 'catch all' element, these transitions were also excluded from the analysis. 4.46% of the tokens in the language sample were not in the first 19,999 most frequent word types in the BNC. Once transitions falling into either of the above categories were excluded from the analysis, 43,853 of the original 50,000 transitions (87.7%) remained in the sample.

Results

The performance of the smoothing methods examined in Simulation 3 is shown in Table 7. The running averages of the per-word perplexity over each transition are plotted in Figure 19 in order to give an idea of how much the estimated transition probability according to each model fluctuates as a function of the nature of the corpus. The estimates appear to have converged fairly reliably after about one third of the test data has been processed, indicating that the test corpus was of a sufficient size in order to permit discriminations between the performance of the various models to be made. As can be seen, the best-performing method was the similarity-based method using the Kneser-Ney covering distribution. This model performed significantly better than the next best method, which was the Modified Kneser-Ney method ($t_{43852} = 2.85$, $p = 0.0043$). In general, the pattern of results revealed a strong superiority for methods based on the Kneser-Ney distribution instead of the unigram distribution: i.e. the Kneser-Ney model outperformed the Absolute Discounting model ($t_{43852} = 24.75$, $p \approx 0$), and the Kneser-Ney version of the similarity-based model outperformed the unigram version ($t_{43852} = 28.42$, $p \approx 0$).

Although the difference between the similarity-based model using the Kneser-Ney covering distribution and the Modified Kneser-Ney model is significant, the 0.37% improvement in per-word perplexity is a modest one (for example, in Simulation 2 the similarity-based Kneser-Ney model improved on the performance of Modified Kneser-Ney by 4.78% in terms of KL distance). We hypothesized that the modest performance improvement could be the result of two factors: (1) that the amount of training data used in Simulation 3 – more than 100 million words, which is approximately twice as many words as a child of age 12 will typically have encountered (Landauer and Dumais, 1997) – is a relatively large amount of training data to use in order to train bigram

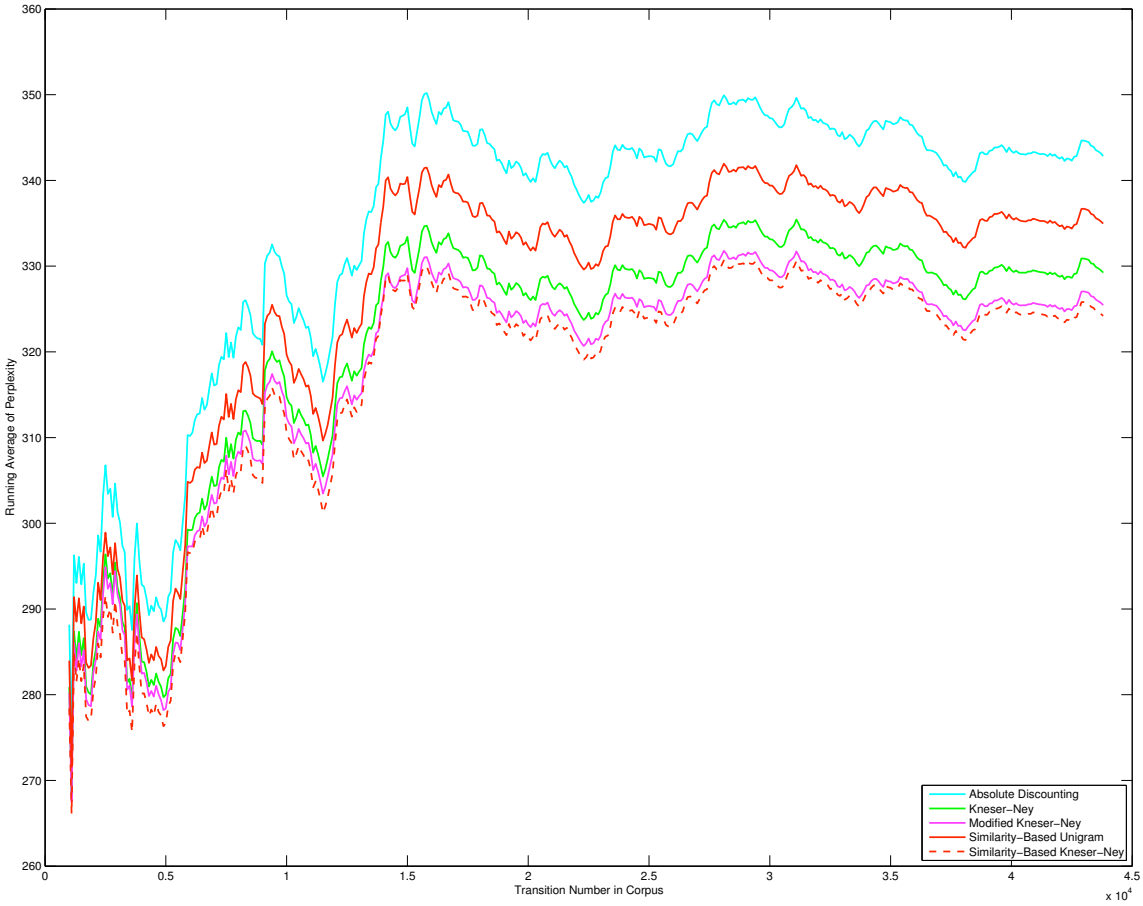

Figure 19. The running average of perplexity in Simulation 3 for each model throughout the language sample.

language models, and that this could have minimized the general impact of data-sparsity in the simulation; and (2) that the majority of words encountered in the language sample would have been high frequency ones as a consequence of Zipf's Law, and for these transitions the difference in smoothing techniques will be minimal because the raw frequency distribution will be the prime determinant of the probability estimates, and these will be the same no matter which smoothing technique is being considered (this would also explain why the difference in performance in Simulations 1 and 2 appeared greater, because here the sampled words were not skewed towards high frequency words as those in a sample of real language will be). This means that the simple mean per-word perplexity statistic could have understated the differences in performance between the various smoothing models.

In order to explore these predictions we grouped the transitions in our sample into 10 bins based on the log frequency of the initial word of each bigram. Figure 20 shows the mean perplexity of transitions falling in each bin for both the Modified Kneser-Ney and the similarity-based Kneser-Ney models (the two best performing models). The percentage improvement gains for the lower frequency transitions are much larger than the overall percentage improvement statistic suggests. This can be seen clearly in Figure 21 in which the percentage improvement in per-word perplexity for the similarity-based Kneser-Ney method over the Modified Kneser-Ney method is plotted for each logarithmic frequency bin. For the lowest frequency transitions (where the bin center implies that the first word of each bigram will only have been seen, on average, 2.18 times), the impact of data-sparsity will be at its most severe and the similarity-based model offers a more than 25% advantage in terms of perplexity over the Modified Kneser-Ney model. Figure 21 therefore provides a much more representative depiction of the advantage of using a similarity-based framework when it comes to language modeling, and supports our hypotheses about the obscured difference.

Turning back to Figure 20, another clear trend for the two models – and indeed, all the models we examined – is the U-shaped per-word perplexity curve as a function of frequency. This curve suggests that it is hard to estimate the probability of the next word when the previous word is either low or high frequency, whereas it is relatively easy to predict the next word when the previous one is of middling frequency. This may appear as a somewhat counterintuitive result. The difficulty of predicting the next word in an infrequent context is easy to explain given our discussion of data-sparsity: the context has been observed so few times that any probability estimates, even smoothed ones, will tend to be unreliable. We believe that the difficulty of predicting the next word when the previous one is high frequency is due to the large preponderance of function words amongst the class of high frequency words. This class of words are ubiquitous in language and are highly promiscuous, in the sense that they can be conjoined with many other words (for example, the determiner 'the' can be followed by any noun, for example, and so predicting exactly which one will occur is a difficult job). Therefore predictions in the mid-frequency range tend to be cases in which data-sparsity is not too severe, but in which the word in question still exerts fairly strong constraints on the next word that can occur).

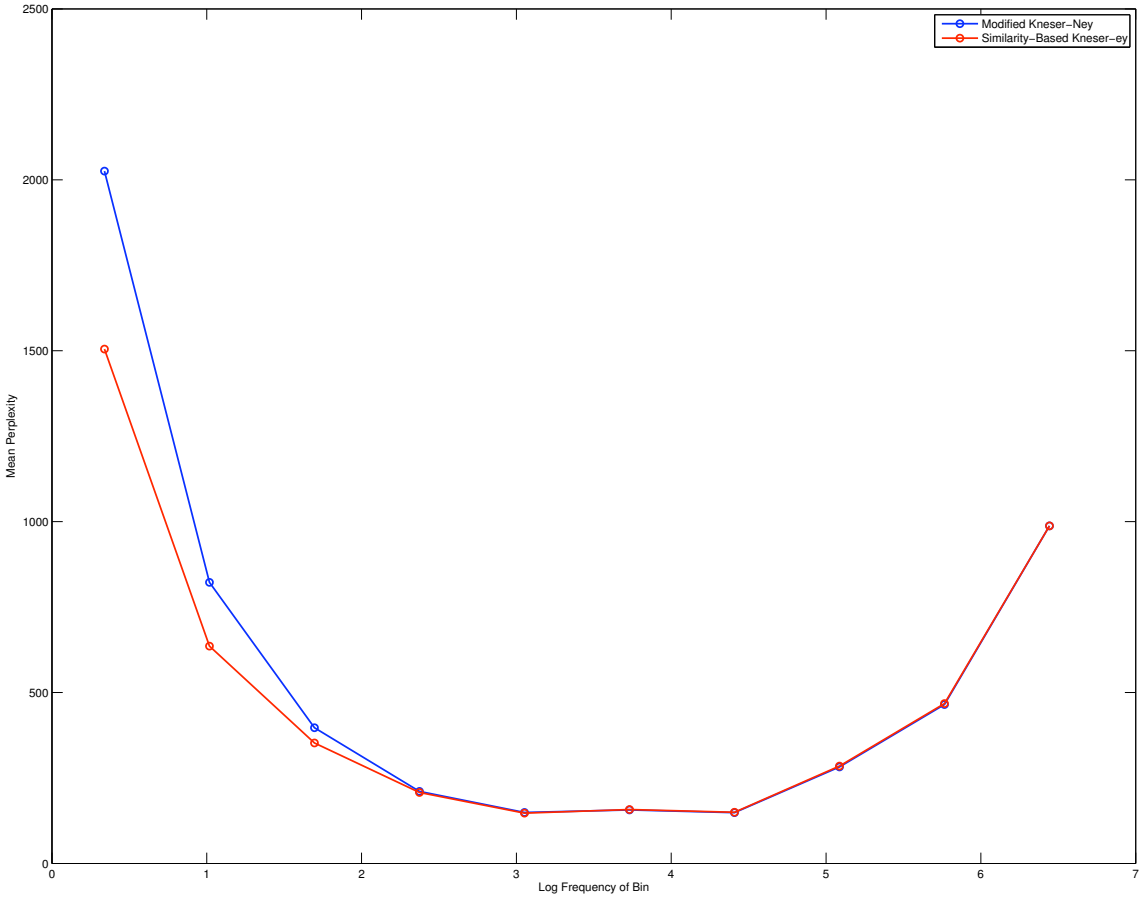

Figure 20. The mean per-word perplexity of transitions for the Modified Kneser-Ney and similarity-based Kneser-Ney model, binned by the log frequency of the initial element of each bigram.

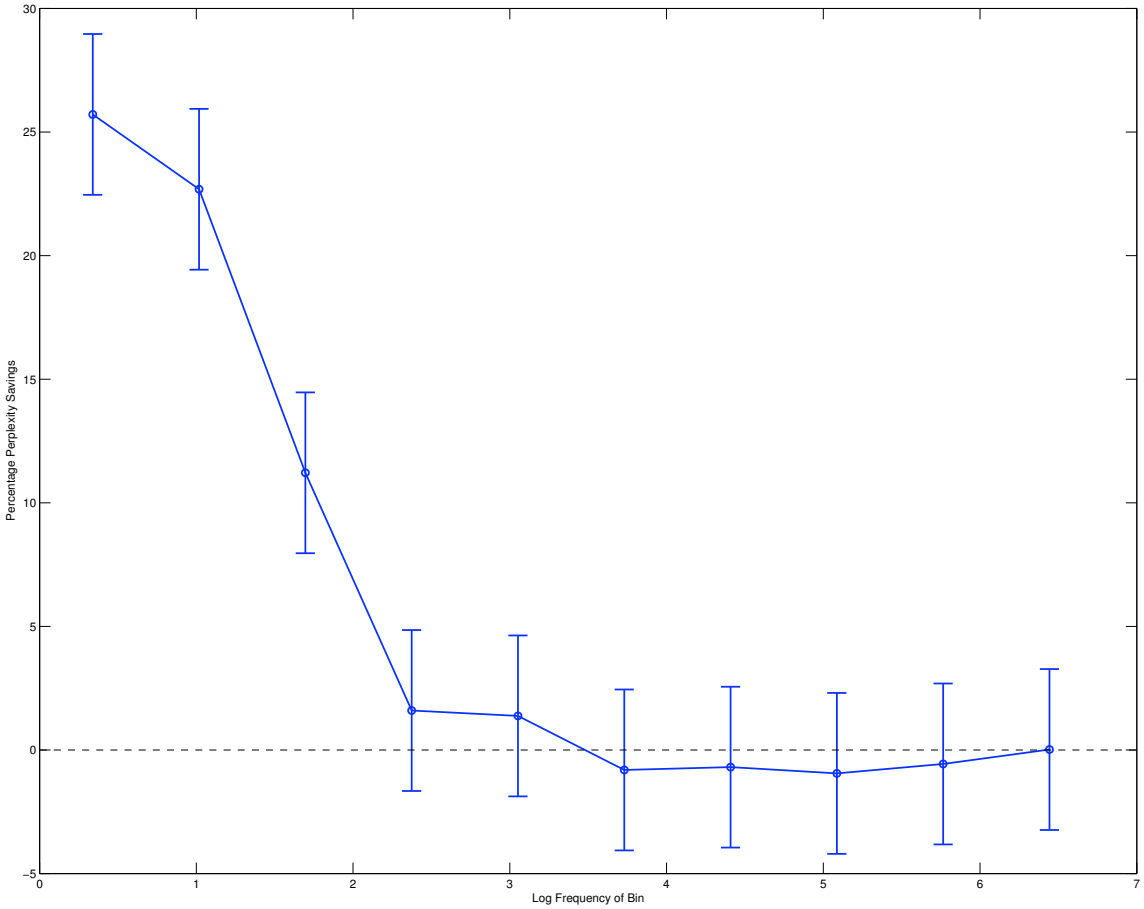

Figure 21. The percentage savings in mean per-word perplexity for the similarity-based Kneser-Ney model over the Modified Kneser-Ney model for bigram transitions binned by the log frequency of their initial element. The similarity-based model produces substantial savings in terms of perplexity for the lower frequency bigrams.

Summary and Discussion

In Simulation 3 we saw that a similarity-based model can outperform existing models for estimating the transitional probabilities in language. Although the overall superiority of the similarity-based Kneser-Ney model over the next best method, Modified Kneser-Ney, was modest, we showed that differences in performance between the two models were obscured by a relatively large training corpus, and by the skewed presence of high frequency transitions in typical language samples. This observation makes an interesting methodological point given the widespread use of per-word perplexity (or one of its cognates) as the measure of performance in language modeling tasks. However, once we analyzed the difference in performance of the models by the frequency of the conditioning word the superiority of the similarity-based model in situations in which data-sparsity is most serious became apparent. In addition, the success of the similarity-based framework in a conventional language modeling task validated the results and methodology of Simulations 1 and 2, and also provided support for our basic thesis that employing similarity-based information could be a beneficial strategy for human language learners.

Future Directions

There are a number of directions in which the framework we have outlined could be extended in future research. Perhaps the most obvious is the application of similarity-based approaches to higher-order language models (i.e. ones which base their prediction of the next word on the previous n words, rather than just the single previous word). As Miller and Chomsky (1963) indicated, any probabilistic model which is to make a serious claim to modeling more than the rudiments of human syntactic competence will need to condition its lexical predictions on a fairly extensive history of words, but it is exactly in this situation that the problem of data-sparsity will be most serious. What is required in order to use a similarity-based model to train an ngram language model is a similarity metric which can operate over sequences of words (of length $n - 1$), and not just individual words. To our knowledge, no similarity metric for word *sequences*, which relies only on information contained in the language stream has been developed to date.

The obvious approach would be to use a distributional method such as the one we have used in our similarity-based model, but where the distributions are defined around sequences of words instead of individual words. While this method should work in principle, it could itself be derailed by data-sparsity: the frequency of a given bigram will tend to be much lower than the frequency of individual words, and hence the distributions observed to occur before and after bigrams will be sparsely populated and hence potentially unreliable (obviously this problem will only get worse the longer the sequences in question become). However, whether the distributional data-sparsity is so severe as to mitigate all of the potential advantages of a similarity-based framework for higher-order language models is an empirical question that deserves investigation. The same objection could have been levied against the similarity-based model presented here: that it is impossible to smooth distributions using a distributional metric, because the distributional metric will be as unreliable as the distributions which we are attempting to smooth. Clearly this objection proved

to be unfounded in the case of bigram language modeling, and so it may also be unfounded in the case of higher-order language models. In any event, the issue deserves further investigation.

A second issue concerning our present results concerns the importance of using a covering distribution. We found that combining the similarity-derived neighboring distribution with the Kneser-Ney distribution in approximately a 4:3 ratio seemed to provide the best smoothing performance in general. However, Dagan, Lee and Pereira (1999) found that the covering distribution seemed to require a lower weighting in their simulations – they found that a 4:1 ratio yielded good performance. However, they did not exhaustively explore all parameter combinations in their study, and it is possible that this could represent a suboptimal selection of parameter values. One difference between our simulations and those reported by Dagan, Lee and Pereira was that their model summed over a greater number of neighboring distributions (their best performing language model summed over the 60 nearest neighboring distributions), and this could have contributed to the difference. When more neighboring distributions are used, the chance of missing an important co-occurrence is reduced, and hence the need for a separate covering distribution may be minimized. Furthermore, it is worth noting that for reasons of simplicity we decided to use the k nearest neighbor variant of the similarity-based framework in Simulation 2, although the power-decay variant actually demonstrated the best smoothing performance in Simulation 1. This model choice could have repercussions on the relative importance of the covering distribution during similarity-based smoothing: because the power-decay has a longer tail, it is less likely to miss significant occurrences when it integrates across the neighboring distributions, and therefore the importance of the covering distribution could be diminished. Indeed, this is supported by an analysis of the optimal parameter values assigned to p_2 in Simulation 1, which controls the weight of the covering distribution. The mean value (and standard errors) of p_2 for the three types of similarity-based model was as follows: power decay, 0.6836 (0.0114), exponential-decay, 0.7800 (0.0101), and k nearest neighbors, 0.7060 (0.0102). As can be seen, the power-decay model required the lowest value of p_2 on average, and this difference was statistically reliable ($F_{2,3599} = 22.61$, $p = 1.75 \times 10^{-10}$). Therefore, in the future further investigation of the importance (and/or necessity) of a covering distribution is merited.

Finally, another avenue for research lies in experimentally testing for effects of similarity-based generalization in a linguistic prediction task. Previously, McDonald and Ramscar (2001) have shown that manipulating the distributional properties of both nonce and unfamiliar words (like *samovar*) does have an effect on the meaning that people attribute to these terms. The next step would be to show that this distributional manipulation also has an impact on the predictions people are willing to make based on these terms. One can envisage designing an artificial language in which a word A is made distributionally similar to either B_1 or B_2, where B_1 and B_2 predict different elements in a language stream (say C_1 and C_2 respectively). The critical test would be to see whether, despite the fact that A has never been paired with C_1, whether people will nevertheless predict C_1 to follow A when it has been made distributionally similar to B_1 (and to predict C_2 when A has been made distributionally similar to B_2). Such a result would provide compelling evidence for the use of a similarity-based mechanism in language learning that we have been arguing for in this paper.

General Discussion

In this article we have been concerned with the possibility of reliably estimating the probabilities of events in language and, more particularly, with the problem of estimating *transitional* or *bigram* probabilities. We began by reviewing theoretical objections raised against the possibility of probabilistic approaches to language, which typically revolve around the notion of data-sparsity in one guise or another, and also presented empirical data attesting to the severity of data-sparsity even in the relatively simple case of attempting to estimate transitional probabilities (in Simulation 1). We pointed out that perhaps the most widely cited argument against the possibility of probabilistic approaches to language, that presented by Miller and Chomsky (1963), assumes that learners utilize an extremely naive method for estimating probabilities – namely the method of maximum likelihood – which is unable to generalize across non-identical word sequences, with the consequence that this method of probability estimation fails to make an efficient use of evidence (linguistic input) that it is provided with. We therefore reviewed alternative approaches to probability estimation that have been proposed in the field of natural language processing, and in addition proposed a similarity-based framework for probability estimation, building on the work of Dagan, Lee and Pereira (1999). In a series of 3 simulations we showed that similarity-based information can be an extremely valuable source of information when it comes to estimating the transitional probability of an event, so much so, in fact, that the similarity-based framework we presented was reliably able to outperform what is generally regarded as the state of the art smoothing technique, the Modified Kneser-Ney method (Chen and Goodman, 1998).

Given the ubiquity of cognitive processes that appear to involve similarity-based generalization (see Hahn and Ramscar, 2001), we believe these results lend credence to the idea that human language learners – and in particular children, who are most strongly afflicted by data-sparsity – could exploit similarity-based information in order to generalize meaningfully from previous linguistic experience. What we hope to have shown in this article is not only that similarity-based information is extremely useful in combatting data-sparsity, but also that the relevant similarities can be extracted from simple co-occurrence statistics which there is good evidence to suggest even children are able to reliably track (Saffran, Aslin and Newport, 1996; Aslin, Saffran and Newport, 1998; Kirkham, Slemmer and Johnson, 2002), which have been implicated in the child's acquisition of part-of-speech categories (Cartwright and Brent, 1997; Redington, Chater and Finch, 1998; Mintz, 2003), and which are able to capture many aspects of semantic knowledge without the need for hand-coded representations (Landauer and Dumais, 1997; Burgess and Lund, 1997).

We also believe our approach is made more plausible in terms of its relevance to human learners by its similarity to another learning mechanism which has been influential in our understanding of language acquisition, the *simple recurrent network* (SRN; Elman, 1990, 1991). SRNs can be trained to make predictions about the next 'word' in a sequence based on preceding elements by using a learning rule which minimizes prediction error on a training set (they can also be applied, for example, to learning transition sequences in a key-pressing task; see Cleeremans and McClelland, 1991). Some of the reasons why SRNs have attracted so much attention in the past is that they are able to learn about the sequential structure of a stimulus field based only on exposure to it, that

they seem to generalize in meaningful ways from particular examples, and their predictions naturally come to depend on multiple preceding elements in a sequence where it is necessary to do so in order to make correct predictions (Cleeremans, Servan-Schreiber and McClelland,1989; Servan-Schreiber, Cleeremans and McClelland, 1991). The similarity-based framework we have presented is different from the class of SRNs in many respects. For example, our model does not rely on an explicit error-reduction strategy as SRNs do, and the framework we have presented is much more scalable than SRNs, that typically require a number of connections which grows exponentially with the size of the vocabulary employed, and suffer from problems of extremely long learning times with complex stimuli. However, one respect in which the two classes of model are similar lies in the way they generalize from previously encountered examples: it has been shown that an SRN develops an internal representation of stimuli which clusters together contexts which tend to lead to similar outcomes, and generalizes across these situations when making its predictions (they can thus be thought of as *graded state machines* which blend together information from similar cues in order to make predictions; see Servan-Schreiber, Cleeremans and McClelland, 1991). This is analogous to the process of summing across a series of neighboring distributions in order to derive a representation of the neighboring distribution which our similarity-based framework relies upon. Indeed, Elman (1990) showed that elementary knowledge about the grammatical classes of words – exactly the same type of information that we derive from our distributional model – seemed to be acquired by an SRN merely from attempting to predict words in a sequence obeying a simple grammatical structure.

While we have only been concerned with the estimation of bigram probabilities in the simulations we have reported, and therefore have not answered Miller and Chomsky's challenge in the general case, we have shown that the impact of data-sparsity for bigrams can be significantly reduced through the use of similarity-based information, and we see no reason why a similar advantage will not be found for similarity-based methods when applied to higher-order language models. Indeed, it may even be that just as the problem of data-sparsity becomes more serious the higher the order of the language model, so the potential benefits from exploiting similarity-based information become all the greater.

References

Aslin R.N., Saffran J.N. and Newport E.L. (1998). Computation of Conditional Probability Statistics by 8-Month-Old Infants, *Psychological Science*, **9**, 321-324.

Baayen R.H. (2001). *Word Frequency Distributions*. Dordrecht: Kluwer Academic Publishers.

Bahl L.R., Jelinek F. and Mercer R.L. (1983). A Maximum Likelihood Approach to Continuous Speech Recognition, *IEEE Journal of Pattern Analysis and Machine Intelligence*, **5**(2), 179-190.

Berko J. (1958). The Child's Learning of English Morphology, *Word*, **14**, 150-177.

Brown P.F., Della Pietra V.J., deSouza P.V., Lai J.C. and Mercer R.L. (1992). Class-Based n-gram Models of Natural Language, *Computational Linguistics*, **18**(4), 467-479.

Burgess C. and Lund K. (1997). Modeling Parsing Constraints with High-Dimensional Context Space, *Language and Cognitive Processes*, **12**(2-3), 177-210.

Burnage G. and Dunlop D. (1992). Encoding the British National Corpus. In J.M. Aarts, P. de Haan and N. Oostdijk (Eds.), *English Language Corpora: Design, Analysis, Exploitation* (pp. 79-95). Amsterdam: Rodopi.

Burnard L. (2000; Ed.). *Reference Guide to the British National Corpus (World Edition)*.

Cartwright and Brent (1997). Syntactic Categorization in Early Language Acquisition: Formalizing the Role of Distributional Analysis, *Cognition*, **63**, 121-170.

Charniak E. (1993). *Statistical Language Learning*. Cambridge, MA: The MIT Press.

Chater N. and Manning C.D. (2006). Probabilistic Models of Language Processing and Acquisition, *Trends in Cognitive Sciences Special Issue: Probabilistic Models of Cognition*, **10**(7), 335-344.

Chen S.F. and Goodman J. (1996). An Empirical Study of Smoothing Techniques for Language Modeling. In *34th Annual Meeting of the ACL* (pp. 310-318). Somerset, New Jersey: Association for Computational Linguistics. (Distributed by Morgan Kaufmann, San Francisco.)

Chen S.F. and Goodman J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling, Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA.

Chomsky N. (1957). *Syntactic Structures*. Mouton.

Chomsky N. (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: The MIT Press.

Chomsky N. (1986). *Knowledge of Language: Its Nature, Origin, and Use.* New York: Praeger.

Church K.W. and 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.

Cleeremans A., Servan-Schreiber D. and McClelland J.L. (1989). Finite State Automata and Simple Recurrent Networks, *Neural Computation*, **1**, 372-381.

Cleeremans A. and McClelland J.L. (1991). Learning the Structure of Event Sequences, *Journal of Experimental Psychology: General*, **120**(3), 235-253.

Culicover P. (1999). *Syntactic Nuts: Hard Cases, Syntactic Theory, and Language Acquisition*. Oxford: Oxford University Press.

Dagan I., Lee L. and Pereira F.C.N. (1999). Similarity-Based Models of Word Cooccurrence Probabilities, *Machine Learning*, **34**, 43-69.

Duda R.O., Hart P.E. and Stork D.G. (2001). *Pattern Classification* (2nd Edition). New York: John Wiley & Sons, Inc.

Efron B. and Thisted R. (1976). Estimating the Number of Unseen Species: How Many Words did Shakespeare Know? *Biometrika*, **63**(3), 435-447.

Elman J.L. (1990). Finding Structure in Time, *Cognitive Science*, **14**, 179-211.

Elman J.L. (1991). Distributed Representations, Simple Recurrent Networks, and Grammatical Structure, *Machine Learning*, **7**, 195-224.

Essen U. and Steinbiss V. (1992). Co-occurrence Smoothing for Stochastic Language Modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Volume 1, pp. 161-164.

Galassi M. Davies J., Theiler J., Gough B., Jungman G., Booth M. and Rossi F. (2005). *GNU Scientific Library Reference Manual, Revised Second Edition*. Bristol, UK: Network Theory Ltd.

Gale W. and Church K. (1994). What's Wrong with Adding One? In N. Oostdijk and P. de Haan (Eds.), *Corpus-Based Research into Languge: In honour of Jan Aarts* (pp.189-200). Rodopi: Amsterdam.

Gale W.A. (1995). Good-Turing Smoothing Without Tears, *Journal of Quantitative Linguistics*, **2**, 217-237.

Gentner D. and Markman A.B. (1997). Structure Mapping in Analogy and Similarity, *American Psychologist*, **52**(1), 45-56.

Gold E.M. (1967). Language Identification in the Limit, *Information and Control*, **16**, 447474.

Goldberg A.E. (2003). Constructions: A New Theoretical Approach to Language, *Trends in Cognitive Sciences*, **7**(5), 219-224.

Good I.J. (1953). The Population Frequencies of Species and the Estimation of Population Parameters, *Biometrika*, **40**(3/4), 237-264.

Good I.J. and Toulmin G.H. (1956). The Number of New Species, and the Increase in Population Coverage, When a Sample is Increased, *Biometrika*, **43**(1-2), 45-63.

Hahn U. and Ramscar M.J.A. (Eds., 2001). *Similarity and Categorisation*. Oxford: Oxford University Press.

Holyoak K.J. and Koh K. (1987). Surface and Structural Similarity in Analogical Transfer, *Memory and Cognition*, **15**, 332-340.

Jeffreys H. (1948). *Theory of Probability* (2^{nd} Edition). Clarendon Press, Oxford.

Jelinek F. (1998). *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press.

Jelinek F. and Mercer R.L. (1980). Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam: North-Holland.

Johnson W.E. (1932). Probability: Deductive and Inductive Problems, *Mind*, **41**, 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*, **ASSP-35**(*3*), 400-401.

Kirkham N.Z., Slemmer J.A. and Johnson S.P. (2002). Visual Statistical Learning in Infancy: Evidence of a Domain General Learning Mechanism, *Cognition*, **83**(2), B35-B42.

Kneser R. and Ney H. (1995). Improved Backing-Off for M-Gram Language Modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 1*, Pages 181-184.

Kullback S. and Leibler R. A. (1951). On Information and Sufficiency, *Annals of Mathematical Statistics*, **22**, 79-86.

Landauer T.K. and Dumais S.T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation, *Psychological Review*, **104**(2), 211-240.

Lee L. (1999). Measures of Distributional Similarity, *37th Annual Meeting of the Association for Computational Linguistics*, 25-32.

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.

MacWhinney B. (2000). *The CHILDES Project: Tools for Analyzing Talk*. Mahwah, NJ: Lawrence Erlbaum Associates.

Manning C.D. and Schütze H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: The MIT Press.

Maslen R.J., Theakston A.L., Lieven E.V. and Tomasello M. (2004). A Dense Corpus Study of Past Tense and Plural Overregularization in English, *Journal of Speech, Language and Hearing Research*, **47**(6), 1319-1333.

McDonald S. and Ramscar M.J.A. (2001). Testing the Distributional Hypothesis: The Influence of Context on Judgments of Semantic Similarity. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*. Edinburgh: University of Edinburgh.

Miller G.A. and Chomsky N. (1963). Finitary Models of Language Users. In Luce, Bush and Galanter (Eds.), *Handbook of Mathematical Psychology, Volume 2*, New York: Wiley (pp. 419-491).

Mintz (2003). Frequent Frames as a Cue for Grammatical Categories in Child Directed Speech, *Cognition*, **90**, 91-117.

Nádas A. (1985). On Turing's Formula for Word Probabilities, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-33**(6), 1414-1416.

Ney H., Essen U. and Kneser R. (1994). On Structuring the Probabilistic Dependencies in Stochastic Language Modeling, *Computer, Speech, and Language*, **8**, 1-38.

Ney H., Essen U., and Kneser R. (1995). On the Estimation of 'Small' Probabilities by Leaving-One-Out, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(12), 1202-1212.

Nosofsky R.M. (1986). Attention, Similarity, and the Identification-Categorization Relationship, *Journal of Experimental Psychology: General*, **115**(1), 39-57.

Pinker S. (1984). *Language Learnability and Language Development*. Cambridge, MA: Harvard University Press.

Pinker S. (1994). *The Language Instinct: How the Mind Creates Language*, New York, NY: William Morrow and Co.

Pinker S. and Prince A. (1988). On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition, *Cognition*, **28**, 73-193.

Pullum G.K. and Scholz B.C. (2002). Empirical Assessment of Stimulus Poverty Arguments, *The Linguistic Review*, **19**, 9-50.

Ramscar M.J.A. and Yarlett D.G. (2007). Linguistic Self-Correction in the Absence Feedback: A New Approach to the Problem of Language Acquisition, *Cognitive Science*, **31**, 1-34.

Redington M., Chater N. and Finch S. (1998). Distributional Information: A Powerful Cue for Acquiring Syntactic Categories, *Cognitive Science*, **22**(4), 425-469.

Rumelhard D.E. and McClelland J.L. (1986). On Learning the Past Tenses of English Verbs. In Rumelhart, McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing, Volume 1*. Cambridge, MA: MIT Press.

Saffran J.R., Aslin R.N. and Newport E.L. (1996). Statistical Learning by 8-Month-Old Infants, *Science*, **274**, 1926-1928.

Saffran J.R., Newport E.L. and Aslin R.N. (1996). Word Segmentation: The Role of Distributional Cues, *Journal of Memory and Language*, **35**, 606-621.

Seidenberg M. S. and MacDonald M. C. (1999). A Probabilistic Constraints Approach to Language Acquisition and Processing, *Cognitive Science*, **23**, 569-588.

Servan-Schreiber D., Cleeremans A. and McClelland J.L. (1991). Graded State Machines: The Representation of Temporal Contingencies in Simple Recurrent Networks, *Machine Learning*, **7**, 161-193.

Shepard R.N. (1987). Towards a Universal Law of Generalization for Psychological Science, *Science*, **237**(4820), 1317-1323.

Skinner B.F. (1957). *Verbal Behavior*. New York: Appleton-Century-Crofts.

Simon H.A. (1955). On a Class of Skew Distribution Functions, *Biometrika*, **42**(3/4), 435-440.

Sloman S. (1993). Feature-Based Induction, *Cognitive Psychology*, **25**, 231-280.

Stemberger J.P. and MacWhinney B. (1988). Are Inflected Forms Stored in the Lexicon? In M. Hammond and M. Noonan (Eds.), *Theoretical Morphology: Approaches in Modern Linguistics* (pp. 101-116). San Diego, CA: Academic Press.

Tomasello M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Cambridge, MA: Harvard University Press.

Tomasello M. and Stahl D. (2004). Sampling Children's Spontaneous Speech: How Much is Enough?, *Journal of Child Language*, **31**, 101-121.

Tversky A. (1977). Features of Similarity, *Psychological Review*, **84**(4), 327-352.

Witten I.H. and Bell T.C. (1991). The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression, *IEEE Transactions on Information Theory*, **37**(4), 1085-1094.

Yamamoto M. and Church K. (2001). Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus, *Computational Linguistics*, **27**(1), 1-30.

Zipf G.K. (1935). *The Psychobiology of Language*. New York, NY: Houghton-Mifflin.

Zipf G.K. (1949). *Human Behavior and the Principle of Least-Effort*. Cambridge, MA: Addison-Wesley.

Appendix
Deriving Good-Turing

The Good-Turing method seeks to get a more reliable estimate of the relative frequency of an item, correcting for the fact that the frequencies observed in a given sample may be biased because of undersampling (data-sparsity). It asks what the expected frequency of an item would be in a second hypothetical sample, given that it had a frequency of r in an observed sample of N items (where the second sample is assumed to be of the same size as the first). In other words, we wish to find the expected frequency of a generic item σ in this second sample, which we will denote r^*, where all we know about σ is that it had a frequency of r in the observed sample:

$$r^* \equiv E\left[R_\sigma^2 | R_\sigma^1 = r\right] \tag{46}$$

where R_σ^i is a random variable denoting the frequency of item σ in the ith sample. Given that our sample is of size N, we can rewrite this as

$$= NE\left[p_\sigma | R_\sigma^1 = r\right] \tag{47}$$

where p_σ denotes the (unknown) probability of an item σ. Assuming that the population being sampled from consists of $|V|$ distinct types of item, we can rewrite the above equation in terms of the standard definition of expectation:

$$= N \sum_{v=1}^{|V|} p_v \cdot P(\text{Item } v \text{ is } \sigma | R_\sigma^1 = r) \tag{48}$$

Now, the probability that item v is σ is simply the probability that item v has a frequency of r divided by the total probability that each has of having a frequency of r (because σ is a generic item, about which the only information we have is that it was observed r times):

$$= \frac{N \sum_{v=1}^{|V|} p_v P(f(I_v) = r)}{\sum_{t=1}^{|V|} P(f(I_t) = r)} \tag{49}$$

Assuming that items occurrences are independently distributed (this assumption is less true when we are concerned with language over other domains, but nevertheless Good-Turing estimators have been usefully applied in language models), we can write the probability of an item having

a frequency of r in terms of the binomial distribution:

$$\begin{aligned} &= \frac{N \sum_{v=1}^{|V|} p_v \binom{N}{r} p_v^r (1-p_v)^{N-r}}{\sum_{t=1}^{|V|} \binom{N}{r} p_t^r (1-p_t)^{N-r}} \\ &= \frac{N \sum_{v=1}^{|V|} p_v^{r+1} (1-p_v)^{N-r}}{\sum_{t=1}^{|V|} p_t^r (1-p_t)^{N-r}} \end{aligned} \tag{50}$$

Inspection shows that the numerator and denominator of the above expression are closely related to the number of item types with a given frequency: the expected number of item types with a frequency of r in our sample is $\binom{N}{r} \sum_{v=1}^{|V|} p_v^r (1-p_v)^{N-r}$. Therefore, if we define $E_N(n_r)$ to mean the expected number of items in a sample of size N to have a frequency of r, the previous equation can be rewritten as:

$$\begin{aligned} &= N \frac{E_{N+1}(n_{r+1}) / \binom{N+1}{r+1}}{E_N(n_r) / \binom{N}{r}} \\ &= N \frac{\binom{N}{r} E_{N+1}(n_{r+1})}{\binom{N+1}{r+1} E_N(n_r)} \\ &= \left(\frac{r+1}{1+1/N} \right) \left(\frac{E_{N+1}(n_{r+1})}{E_N(n_r)} \right) \end{aligned} \tag{51}$$

Now, when N is large (for example, around 10^7 or more, as it is usually in language modeling applications), the following approximation holds:

$$\frac{r+1}{1+1/N} \approx r+1 \tag{52}$$

Substituting Equation 52 into Equation 51 implies,

$$r^* \approx (r+1) \frac{E_{N+1}(n_{r+1})}{E_N(n_r)} \tag{53}$$

which completes the derivation. For alternative derivations of the basic Good-Turing result see Good (1953), Church and Gale (Appendix A, 1991), Nádas (1985) or Baayen (2001).