Similarity-Based Approaches to Natural Language Processing

A thesis presented
by

Lillian Jane Lee

to
The Division of Engineering and Applied Sciences
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer Science

Harvard University
Cambridge, Massachusetts

May 1997

Abstract

Statistical methods for automatically extracting information about associations between words or documents from large collections of text have the potential to have considerable impact in a number of areas, such as information retrieval and natural-language-based user interfaces. However, even huge bodies of text yield highly unreliable estimates of the probability of relatively common events, and, in fact, perfectly reasonable events may not occur in the training data at all. This is known as the *sparse data problem.* Traditional approaches to the sparse data problem use crude approximations. We propose a different solution: if we are able to organize the data into classes of similar events, then, if information about an event is lacking, we can estimate its behavior from information about similar events. This thesis presents two such similarity-based approaches, where, in general, we measure similarity by the Kullback-Leibler divergence, an information-theoretic quantity.

Our first approach is to build soft, hierarchical clusters: soft, because each event belongs to each cluster with some probability; hierarchical, because cluster centroids are iteratively split to model finer distinctions. Our clustering method, which uses the technique of deterministic annealing, represents (to our knowledge) the first application of soft clustering to problems in natural language processing. We use this method to cluster words drawn from 44 million words of Associated Press Newswire and 10 million words from Grolier's encyclopedia, and find that language models built from the clusters have substantial predictive power. Our algorithm also extends with no modification to other domains, such as document clustering.

Our second approach is a nearest-neighbor approach: instead of calculating a centroid for each class, we in essence build a cluster around each word. We compare several such nearest-neighbor approaches on a word sense disambiguation task and find that as a whole, their performance is far superior to that of standard methods. In another set of experiments, we show that using estimation techniques based on the nearest-neighbor model enables us to achieve perplexity reductions of more than 20 percent over standard techniques in the prediction of low-frequency events, and statistically significant speech recognition error-rate reduction.

Acknowledgements

In my four years as a graduate student at Harvard, I have noticed a rather ominous trend. Aiken Computation Lab, where I spent so many days and nights, is currently slated for demolition. I worked at AT&T Bell Labs for several summers – and now this institution no longer exists. I finally began to catch on when, after a rash of repetitive strain injury cases broke out at Harvard, one of my fellow graduate students called the Center for Disease Control to suggest that I might be the cause.

All this aside, I feel that I have been incredibly fortunate. First of all, I have had the Dream Team of NLP for my committee. Stuart Shieber, my advisor, has been absolutely terrific. The best way to sum up my interactions with him is: he never let me get away with anything. The stuff I've produced has been the clearer and the better for it. Barbara Grosz has been wonderfully supportive; she also throws a mean brunch. And there is just no way I can thank Fernando Pereira enough. He is the one who started me off on this research enterprise in the first place, and truly deserves his title of mentor.

I'd also like to thank Harry Lewis for the whole CS 121 experience, Les Valiant and Alan Yuille for being on my oral exam committee, and Margo Seltzer for advice and encouragement.

There have been a number of people who made the grad school process much, much easier to deal with: Mike ("at least you're not in medieval history") Bailey, Ellie Baker, Michael Bender, Alan Capil, Stan Chen (the one and only), Ric Crabbe, Adam Deaton, Joshua Goodman (for the colon), Carol Harlow, Fatima Holowinsky, Bree Horwitz, Andy Kehler, Anne Kist, Bobby Kleinberg, David Mazieres, Jeff Miller, Christine Nakatani, Wheeler Ruml, Kathy Ryall (my job buddy), Ben Scarlet, Rocco Servedio (for the citroids), Jen Smith, Nadia Shalaby, Carol Sandstrom and Chris Small (and Emily and Sophie, the best listeners in the entire universe), Peg Schafer, Keith Smith, Chris Thorpe, and Tony Yan.

AT&T was a wonderful place to work, and I am quite grateful for having had the opportunity to talk with all of the following people: Hiyan Alshawi, Ido Dagan, Don Hindle, Julia Hirschberg, Yoram Singer, Tali Tishby, and David Yarowsky. I'd also like to mention the "s underscores" – the summer students at AT&T who made it so difficult to get anything done: David Ahn, Tom Chou, Tabora Constantennia, Charles Isbell, Raj Iyer, Kim Knowles, Nadya Mason, Leah McKissic, Diana Meadows, Andrew Ng, Marianne Shaw, and Ben Slusky.

Rebecca Hwa deserves special mention. She was always there to say "Now just calm down, Lillian" whenever I got stressed out. I thank her for all those trips to Toscanini's, and forgive her for trying to make me break my Aiken rule. Maybe one day I'll even get around to giving her her LaTeX book back.

And finally, deepest thanks to my mom and dad, my sister Charlotte, and Jon Kleinberg; they always believed, even when I didn't.

The work described in this thesis was supported in part by the National Science Foundation under Grant No. IRI-9350192. I also gratefully acknowledge support from an NSF Graduate Fellowship and an grant from the AT&T Labs Fellowship Program (formerly the AT&T Bell Labs Graduate Research Program for Women).

Bibliographic Notes

Portions of this thesis are joint work and have appeared elsewhere.

Chapter 3 is based on the paper "Distributional Clustering of English Words" with Fernando Pereira and Naftali Tishby (Pereira, Tishby, and Lee, 1993), which appeared in the proceedings of the 31st meeting of the ACL. We thank Don Hindle for making available the 1988 Associated Press verb-object data set, the Fidditch parser, and a verb-object structure filter; Mats Rooth for selecting the data set consisting of objects of "fire" and for many discussions; David Yarowsky for help with his stemming and concordancing tools; and Ido Dagan for suggesting ways to test cluster models.

Chapter 4 and portions of chapter 2 are adapted from "Similarity-Based Methods for Word Sense Disambiguation". This paper, co-written with Ido Dagan and Fernando Pereira, will appear in the proceedings of the 35th meeting of the ACL (Dagan, Lee, and Pereira, 1997). We thank Hiyan Alshawi, Joshua Goodman, Rebecca Hwa, Stuart Shieber, and Yoram Singer for many helpful comments and discussions.

Chapter 5 is based on work with Ido Dagan and Fernando Pereira that is described in "Similarity-Based Estimation of Word Cooccurrence Probabilities", which appeared in the proceedings of the 32nd Annual meeting of the ACL (Dagan, Pereira, and Lee, 1994). We thank Slava Katz for discussions on the topic of this paper, Doug McIlroy for detailed comments, Doug Paul for help with his baseline back-off model, and Andre Ljolje and Michael Riley for providing the word lattices for our experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"You shall know a word by the company it keeps!" (Firth, 1957, pg. 11)

We begin by considering the problem of predicting string probabilities. Suppose we are presented with two strings,

1. "Grill doctoral candidates", and
2. "Grill doctoral updates",

and are asked to determine which string is more likely. Notice that this is *not* the same question as asking which of these strings is grammatical. In fact, both constitute legitimate English sentences. The first sentence is a command to ask a graduating Ph.D. student many difficult questions; the second might be an order to take lists of people who have just received doctorates and throw them[1] on a Hibachi.

Methods for assigning probabilities to strings are called *language models*. In this thesis, we will abuse the term somewhat and refer to methods that assign probabilities to word associations as language models, too. That is, we will consider methods which estimate the probability of word cooccurrence relations; these methods need not be defined on sentences. For example, in chapters 3 and 4 we will be concerned with the problem of estimating the probability that a noun x and a transitive verb y appear in a sentence with x being the head noun of the direct object of y.

One important application of language modeling is error correction. Current speech recognizers do not achieve perfect recognition rates, and it is easy to imagine a situation in which a speech recognizer cannot decide whether a speaker said "Grill doctoral candidates" or "Grill doctoral updates". A language model can provide a speech recognizer with the information that the former sentence is more likely than the latter; this information would help the recognizer make the right choice. Similar situations arise in handwriting recognition, spelling correction, optical character recognition, and so on — whenever the physical evidence itself may not be enough to determine the corresponding string.

More formally, let E be some physical evidence, and suppose we wish to know whether the string W is the message conveyed or encoded by E. Using Bayes' rule, we can combine the estimate $P(E|W)$ given by an acoustic model with the probability $P_{LM}(W)$ assigned

[1](the lists)

by a language model to find the posterior probability that W is the true string given the evidence at hand:

$$P(W|E) = \frac{P_{LM}(W)P(E|W)}{P(E)} \tag{1.1}$$

(since the evidence E is fixed, it is the same for every hypothesized string W, so the $P(E)$ term is generally ignored in practice). Thus, in a situation where two hypothesized strings cannot be distinguished on the basis of the physical evidence alone, a language model can provide the information necessary for disambiguation.

Another application of language modeling is machine translation. Suppose one needs to translate the phrase "Grill doctoral candidates" to another language. Two possible target sentences are "Ask applicants many questions" and "Roast applicants on a spit". If we have a language model that furnishes us the information that the first sentence is more likely than the second, then (in the absence of context providing evidence to the contrary) we would pick the first sentence as the correct translation.

This thesis is concerned with *statistical* approaches to problems in natural language processing. Typically, statistical approaches take as input some large sample of text, which may or may not be annotated in some fashion, and attempt to learn characteristics of the language from the statistics in the sample. They may also make use of auxiliary information gained from such sources as on-line dictionaries or WordNet (Miller, 1995). An important advantage of statistical approaches over traditional linguistic models is that statistical methods yield probabilities. These probabilities can easily be combined with estimates from other components, as in equation (1.1) above. Traditional linguistic models, on the other hand, only describe whether or not a string is grammatical. This information is too coarse-grained for use in practical tasks; for instance, both "Grill doctoral candidates" and "Grill doctoral updates" are valid sentences, and yet we know that the first string is far more likely than the second.

Perhaps the simplest statistical approach to language modeling is the maximum likelihood estimate (MLE), which simply counts the number of times that the string of interest occurs in the training sample S and normalizes by the sample size. For "Grill doctoral candidates", this estimate takes the form

$$P_{MLE}(\text{“Grill doctoral candidates”}) = \frac{C(\text{“Grill doctoral candidates”})}{|S|}, \tag{1.2}$$

where $C(\text{“Grill doctoral candidates”})$ is the number of times the phrase occurred in S. The quantity $|S|$ might be the number of word triples in S, or the number of sentences in S, or some other relevant measure.

Notice that if the event of interest is *unseen*, that is, does not occur in S, then the maximum likelihood estimate assigns it a probability of zero. In terms of practicality, this turns out to be a fatal flaw because of the *sparse data* problem: even if S is quite big, a large number of possible events will not appear in S. Assigning all unseen events a probability of zero, as the MLE does, amounts to declaring many perfectly reasonable strings to have zero probability of occurring, which is clearly unsatisfactory.

To illustrate the pernicious nature of the sparse data problem, we present the following example. Consider the set S to be the text contained in all the pages indexed by AltaVista, Digital's web search engine (Digital Equipment Corporation, 1997). Currently, this set consists of 31 million web pages, which, at an extremely conservative estimate, means that

S contains at least a billion words. Yet at the time of this writing, the phrase "Grill doctoral candidates" does not occur at all among those billion words, so the MLE would rule out this sentence as absolutely impossible.

Although the sparse data problem affects low-frequency events, it is incorrect to infer that it therefore is not important. One might attempt to claim that if an event has such a low probability that it does not occur in a very large sample, then actually estimating its probability to be zero will not be a major error. However, the aggregate probability of unseen events can be a big percentage of the test data, which means that it is quite important to treat unseen events carefully. Brown et al. (1992), for instance, studied a 350 million word sample of English text, and estimated that in any new sample drawn from the same source distribution, 14% of the trigrams (sequences of three consecutive words) would not have occurred in the large text. A speech recognizer that refused to accept 1 out of every 7 sentences would be completely unusable.

As a historical aside, we observe that Noam Chomsky famously declared that sparse data problems are insurmountable.

> It is fair to assume that neither sentence (1) [Colorless green ideas sleep furiously] nor (2) [Furiously sleep ideas green colorless] ... has ever occurred .... Hence, in any statistical model ... these sentences will be ruled out on identical grounds as equally 'remote' from English. Yet (1), though nonsensical, is grammatical, while (2) is not."[2] (Chomsky, 1964, pg. 16)

This thought experiment helped "[disabuse] the field once and for all of the notion that there was anything of interest to statistical models of language" (Abney, 1996).

However, in the years since Chomsky wrote this remark, some progress on ameliorating the sparse data problem has been made. Indeed, Chomsky's statement is based on the false assumption that "any statistical model" must be based on the maximum likelihood estimate. This is certainly not the case. Two standard language modeling techniques used in speech recognition, Jelinek-Mercer smoothing and Katz back-off smoothing, make use of an estimator guaranteed to be non-zero. In the case where the probability of an unseen word pair (w_1, w_2) is being estimated, these methods incorporate the probability of word w_2 (details can be found in section 2.2). But this is not always adequate: for example, the word "updates" appears on more web pages indexed by AltaVista than "candidates" does.

The key idea in this thesis is that we can use *similarity* information to make more sophisticated probability estimates when sparse data problems occur. This idea is intuitively appealing, for if we know that the word "candidates" is somehow similar to the word "nominees", then the occurrence of the sentence "Grill doctoral nominees" would lead us to believe that "Grill doctoral candidates" is also likely.

The notion of similarity we explore is that of *distributional similarity*, since we will represent words as distributions over the contexts in which they occur (as implied by the quotation that opens this chapter). We will thus be concerned with measures of the "distance" between probability mass functions. We discuss several such measures in chapter 2, but our main focus will be on using the Kullback-Leibler divergence, an information-theoretic quantity.

[2] Ironically, this remark is now so well known that it has become false: use of AltaVista reveals that at the time of this writing, 40 web pages contain the first sentence, whereas only three contain the second.

The work presented in this thesis can be divided into two parts. The first is the development of a *distributional clustering* method for grouping similar words. This method builds probabilistic, hierarchical clusters: objects belong to each cluster with some probability, and clusters are broken up into subclusters so that a hierarchy results. We derive the method, exhibit clusters found by our method in order to provide a qualitative sense of how the method performs, and show that effective language models can be constructed from the clusters produced by our method. To our knowledge, this is the first probabilistic clustering method to be applied to natural language processing.

The second part is the development of a more computationally efficient way to take incorporate similarity information: a nearest-neighbor (or "most similar neighbor") language model that combines estimates from specific objects rather than from classes. We compare several different implementations of this type of model against standard smoothing methods and find that using similarity information leads to far better estimates.

This thesis is organized as follows. Chapter 2 describes many of the theoretical results employed in later chapters. We discuss standard language modeling techniques and study the properties of several distributional similarity functions. Chapter 3 presents our distributional clustering method. Chapter 4 develops the nearest-neighbor approach and compares the performance of several implementations on a pseudo-word-disambiguation task. Chapter 5 considers an extension of our nearest-neighbor approach and studies its performance on more realistic tasks. We conclude with a brief summary of the thesis and indicate directions for further work in chapter 6.

Chapter 2

Distributional Similarity

This chapter presents background material underlying the work in this thesis. In section 2.1 we argue that representing objects as distributions is natural and useful. Section 2.2 reviews common methods for estimating distributions from a sample. We use these methods to provide initial distributions to our algorithms and also as standards against which to compare the performance of our similarity-based estimates. Section 2.3 studies various functions measuring similarity between distributions. We pay particular attention to the Kullback-Leibler divergence (Cover and Thomas, 1991), which plays a central role in our work.

2.1 Objects as Distributions

The first issue we must address is what representation to use for the objects we wish to cluster and compare. For the moment, we will be vague about what sorts of objects we will be considering; researchers have clustered everything from documents (Salton, 1968; Cutting et al., 1992) to irises (Fisher, 1936; Cheeseman et al., 1988).

We want the representation we choose to satisfy two requirements. First, the representation should be general enough to apply to many different types of objects. Second, any particular object's representation should be easy to calculate from samples alone; we do not want to use outside sources of information such as on-line dictionaries. This second condition expresses our preference for algorithms that are adaptable; if we rely on knowledge that is hard for computers to derive from training data, then we cannot use our algorithms on new domains without expending considerable effort on re-acquiring the requisite knowledge. Furthermore, large samples that have few or no annotations[1] are far more common and readily obtainable than large highly-annotated samples, and thus working with representations adhering to the second condition tends to be much more convenient.

Many clustering schemes represent objects in terms of a set $\{A_1, A_2, \ldots, A_N\}$ of attributes (Kaufman and Rousseeuw, 1990). Each object is associated with an attribute vector $(a_1, a_2, \ldots, a_N)$ of values for the attributes. Some attributes can take on an infinite number of values; for example, the mean of a normal distribution can be any real number. Other attributes, such as the sex of a patient, range only over a finite set. Usually, no assumptions are made about the relationship between different attributes.

[1] In the following chapters, we use either unannotated data or data that has been tagged with parts of speech.

In this thesis, we use a restricted version of the attribute representation. Objects are equated with probability mass functions, or distributions: each attribute must have a non-negative real value, and we require that all object attribute vectors $(a_1, a_2, \ldots, a_N)$ satisfy the constraint that $\sum_{i=1}^{N} a_i = 1$. We can think of a_i as the probability that the object assigns to A_i. This distributional representation for objects is particularly appropriate for situations arising in unsupervised learning, where a learning algorithm must infer properties of events from a sample of unannotated data. In such situations, we can define the attributes to be the contexts in which events can occur; the value a_i for a particular event is then the proportion of the time the event occurred in context i.

For example, suppose we wish to learn about word usage from the following (small) sample of English text: "A rose is a rose is not a nose". Our events are therefore words. If we define the context of a word to be the following word, then the possible contexts are "a", "is","nose", "not", and "rose". The attribute vector for the word "a" is $(0, 0, 1/3, 0, 2/3)$, since "a" occurs before "nose" (the third attribute) one out of three times, and before "rose" (the last attribute) twice.

In accordance with our first requirement for representations, the distributional representation is fairly general. For instance, we have demonstrated that words can be represented as distributions over subsequent words, and we can just as easily represent documents as distributions over the words that occur in them, or customers as distributions over the products they buy, and so on. Indeed, it is a reasonable representation whenever the data consists of a set of events (e.g., words occurring together) rather than measurements or properties (e.g., a list of each word's part of speech). Also, in compliance with our second requirement for distributions, the distributional representation for any object is trivial to calculate as long as the contexts are easily recognizable. Furthermore, we wish to apply our techniques to language modeling, a task for which probability distributions must be produced. Finally, the constraint that the components of attribute vectors sum to unity is of use to us in our calculations, as will be seen in chapter 3.

2.2 Initial Estimates for Distributions

The remainder of this thesis will be concerned with object distributions that have been estimated from object-context pairs. More formally, let $\mathcal{X}$ be the set of objects under consideration and $\mathcal{Y}$ be the set of possible contexts, $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$. Assume that the data consists of pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ along with counts $C(x, y)$ of how many times (x, y) occurred in some training sample. Counts for individual objects and contexts are readily attained from counts for the pairs: $C(x) = \sum_y C(x, y)$ and $C(y) = \sum_x C(x, y)$; without loss of generality, assume that every object and every context occurs at least once. We wish to represent object x by the conditional distribution $P(y|x)$ all $y \in \mathcal{Y}$. This distribution must be estimated from the data pairs. Of course, the goal of this thesis is to develop good estimates for $P(y|x)$, but we need some initial distributions to start with.

A particularly simple estimation method is the *maximum likelihood estimate* (MLE) $P_{MLE}(y|x)$:

$$P_{MLE}(y|x) = \frac{C(x, y)}{C(x)}. \tag{2.1}$$

Notice that if the joint event (x, y) never occurs, then $P_{MLE}(y|x) = 0$, which is equivalent to saying that any event that does not occur in the training sample is impossible. As noted in chapter 1, using the maximum likelihood estimate tends to grossly underestimate the probability of low-frequency events.

Many alternatives to the MLE (Good, 1953; Jelinek and Mercer, 1980; Katz, 1987; Church and Gale, 1991) take the MLE as an initial estimate and adjust it so that the total estimated probability of pairs occurring in the sample is less than one, leaving some probability mass for unseen pairs. These techniques are known as *smoothing* methods, since they "smooth over" zeroes in distributions. Typically, the adjustment involves either *interpolation*, in which the new estimator is a weighted combination of the MLE and an estimator guaranteed to be nonzero for unseen pairs, or *discounting*, in which the MLE is decreased to create some leftover probability mass for unseen pairs.

The work of Jelinek and Mercer (1980) is the classic interpolation method. They produce an estimate by linearly interpolating the MLE for the conditional probability of an object-context pair (x, y) with the maximum likelihood estimate $P_{MLE}(y) = C(y)/\sum_y C(y)$ for the probability of context y:

$$P_{JM}(y|x) = \lambda(x)P_{MLE}(y|x) + (1 - \lambda(x))P_{MLE}(y). \tag{2.2}$$

The function $\lambda(x)$ ranges between 0 and 1, and reflects our confidence in the available data regarding x. If x occurs relatively frequently, then we have reason to believe that the MLE for the pair (x, y) is reliable. We then give $\lambda(x)$ a high value, so that P_{JM} depends mostly on $P_{MLE}(y|x)$. On the other hand, if x is relatively rare, then $P_{MLE}(y|x)$ is unlikely to be very accurate. In this case, we decide to rely more on $P_{MLE}(y)$, since counts for the single event y are higher than counts for the joint event (x, y). We therefore set $\lambda(x)$ to a relatively low value. A method for training λ is described by Bahl, Jelinek, and Mercer (1983).

A popular alternative in the speech recognition literature is the back-off discounting method of Katz (1987). It provides a clear separation between frequent events, for which observed frequencies are reliable probability estimators, and low-frequency events, whose prediction must involve additional information sources. Furthermore, the back-off model does not require complex estimation calculations for interpolation parameters such as $\lambda(x)$, as is the case for Jelinek and Mercer's method.

Katz first uses the Good-Turing formula (Good, 1953) to replace the actual frequency $C(x, y)$ of an object-context pair with a discounted frequency $C^*(x, y)$. Let n_m denote the number of pairs that occurred m times in the sample. The Good-Turing estimate then defines $C^*(x, y)$ as

$$C^*(x, y) = (C(x, y) + 1)\frac{n_{C(x,y)+1}}{n_{C(x,y)}}.$$

This discounted frequency is used in the same way the true frequency is used in the MLE (equation (2.1)):

$$P_d(y|x) = \frac{C^*(x, y)}{C(x)}.$$

As a consequence, the estimated conditional probability of an unseen pair (x', y') is

$$P_d(y'|x') = \frac{(n_1/n_0)}{C(x')}.$$

Thus, the probability mass assigned to unseen pairs involving object x' is distributed uniformly. The total mass assigned to unseen pairs involving x' is simply the complement of the mass assigned to seen pairs involving x':

$$\tilde{\beta}(x') = 1 - \sum_{y:C(x',y)>0} P_d(y|x').$$

For more details, see Nádas (1985), who presents three different derivations (two empirical-Bayesian and one empirical) of the Good-Turing estimate.

Katz alters the Good-Turing treatment by not using P_d for unseen pairs. Rather, he bases his estimate of the conditional probability of an unseen pair (x', y') on an estimate of the probability of y'. This amounts to assuming that the behavior of (x', y') is independent of the behavior of x'; Jelinek and Mercer (1980) make a similar assumption when they set λ to a low value in equation (2.2).

More formally, we write the estimate for an arbitrary pair (x, y) in the following form, which is not Katz's original presentation but will be convenient for us in chapters 4 and 5 (note the asymmetrical treatment of seen and unseen pairs):

$$\hat{P}(y|x) = \begin{cases} P_d(y|x) & \text{if } C(x,y) > 0 \\ \alpha(x) P_r(y|x) & \text{otherwise } ((x,y) \text{ is unseen}) \end{cases} . \tag{2.3}$$

P_r is the model for probability redistribution among unseen pairs. Katz (implicitly) defines P_r as the probability of the context:

$$P_r(y|x) = P(y),$$

so that

$$P_{BO}(y|x) = \begin{cases} P_d(y|x) & \text{if } C(x,y) > 0 \\ \alpha(x) P(y) & \text{otherwise} \end{cases} . \tag{2.4}$$

In later chapters, we will take advantage of the placeholder P_r to insert our own similarity-based probability redistribution models. The quantity $\alpha(x)$ is a normalization factor required to ensure that $\sum_y P_{BO}(y|x) = 1$:

$$\begin{aligned} \alpha(x) &= \frac{\tilde{\beta}(x)}{\sum_{y:C(x,y)=0} P_r(y|x)} \\ &= \frac{\tilde{\beta}(x)}{1 - \sum_{y:C(x,y)>0} P_r(y|x)}. \end{aligned}$$

The second formulation of the normalization is computationally preferable because it is generally the case that the total number of possible pairs far exceeds the number of observed pairs.

Should we use Jelinek-Mercer smoothing, Katz back-off smoothing, or perhaps some other technique? A thorough study by Chen and Goodman (1996) showed that back-off and Jelinek-Mercer smoothing perform consistently well, with back-off generally yielding better results for modeling pairs. Since the back-off formulation also contains a placeholder for us to apply similarity-based estimates, we will use Katz's estimation method whenever smoothed distributions are required.

2.3 Measures of Distributional Similarity

In this section, we consider theoretical and computational properties of several functions measuring the "similarity" between distributions. We refer to these functions as distance functions, rather than similarity functions, since most of them achieve their *minimum* when the two distributions being compared are *maximally* similar (i.e., identical). The work described in chapters 4 and 5 uses negative exponentials of distance functions when true similarity functions (that is, functions that increase as similarity increases) are required.

We certainly do not intend to give an exhaustive listing of all distance functions. (See Anderberg (1973) for an extensive survey.) Our purpose is simply to examine important properties of functions that we use or that are commonly employed by other researchers in natural language processing and machine learning.

We discuss the KL divergence in section 2.3.1 in detail, as it forms the basis for most of the work in this thesis. We also describe several other distance functions, including the total divergence to the mean (section 2.3.2), various geometric norms (section 2.3.3), and some similarity statistics (section 2.3.4). We will pay particular attention to the computational requirements of these functions. In view of the fact that we wish to use very large data sets, we will require that the time needed to calculate the distance between any two distributions be linear or near-linear in the number of attributes. This demand is not strictly necessary for the work described in this thesis – the clustering work of chapter 3 depends on the use of the KL divergence, and the similarity computations of chapters 4 and 5 are done in a preprocessing phrase. However, one of our future goals is to find adaptive versions of our algorithms, in which case we must use functions that can be computed efficiently.

We defer discussion of the *confusion probability*, defined by Essen and Steinbiss (1992), until chapter 4. This function is of great importance to us because Essen and Steinbiss's *co-occurrence smoothing* method is quite similar to our own work on language modeling. The reason we do not include the confusion probability in this chapter is that it is not a function of two distributions : each object x is described both by the conditional probability $P(y|x)$ and the marginal probability $P(x)$, so that comparing two objects involves four distributions.

For the remainder of this section, let x_1, x_2, and x_3 be three objects with associated distributions $P(\cdot|x_1)$, $P(\cdot|x_2)$, and $P(\cdot|x_3)$, respectively. It doesn't matter how these distributions were estimated. For notational convenience, we will call these distributions q, r, and s. We will occasionally refer to a distribution p by its corresponding attribute vector $(p(y_1), p(y_2), \ldots, p(y_N))$.

2.3.1 KL Divergence

We define the function $D(q||r)$ as

$$D(q||r) = \sum_{y \in \mathcal{Y}} q(y) \log \frac{q(y)}{r(y)} \tag{2.5}$$

(we will not specify the base of the logarithm). Limiting arguments lead us to set $0 \log \frac{0}{r} = 0$, even if $r = 0$, and $q \log \frac{q}{0} = \infty$ when q is not zero.

Function (2.5) goes by many names in the literature, including information gain (Rényi, 1970), error (Kerridge, 1961), relative entropy, cross entropy, and Kullback Leibler distance (Cover and Thomas, 1991). Kullback himself refers to the function as information for discrimination, reserving the term "divergence" for the symmetric function $D(q||r)+D(r||q)$ (Kullback, 1959). We will use the name *Kullback-Leibler (KL) divergence* throughout this thesis.

The KL divergence is a standard information-theoretic "measure" of the dissimilarity between two probability mass functions, and has been applied to natural language processing (as described in this thesis), machine learning, and statistical physics. It is not a metric in the technical sense, for it is not symmetric and does not obey the triangle inequality (see, e.g., theorem 12.6.1 of Cover and Thomas (1991)). However, it is non-negative, as shown in the following theorem.

Theorem 2.1 (Information inequality) $D(q||r) \geq 0$, *with equality holding if and only if* $q(y) = r(y)$ *for all* $y \in \mathcal{Y}$.

Proof. Most authors prove this theorem using *Jensen's inequality*, which deals with expectations of convex functions (notice that $D(q||r)$ is the expected value with respect to q of the quantity $\log(q/r)$). However, we present here a short proof attributed to Elizabeth Thompson (Green, 1996).

Let ln denote the natural logarithm, and let $b > 0$ be the base of the logarithm in (2.5). First observe that for any $z \geq 0$, $\ln(z) \leq z - 1$, with equality holding if and only if $z = 1$. Then, we can write

$$\begin{aligned} -D(q||r) &= \sum_{y \in \mathcal{Y}} q(y) \log_b \frac{r(y)}{q(y)} \\ &= \frac{1}{\ln(b)} \sum_{y \in \mathcal{Y}} q(y) \ln \frac{r(y)}{q(y)} \\ &\leq \frac{1}{\ln(b)} \sum_{y \in \mathcal{Y}} q(y) \left(\frac{r(y)}{q(y)} - 1 \right) \\ &= \frac{1}{\ln(b)} \left(\sum_{y \in \mathcal{Y}} r(y) - \sum_{y \in \mathcal{Y}} q(y) \right) \\ &= \frac{1}{\ln(b)} (1 - 1) = 0, \end{aligned}$$

with equality holding if and only if $\frac{r(y)}{q(y)} = 1$ for all $y \in \mathcal{Y}$. ∎

Since the KL divergence is 0 when the two distributions are exactly the same and greater than 0 otherwise, it is really a measure of dissimilarity, as mentioned above, rather than similarity. This yields an intuitive explanation of why we should not expect the KL divergence to obey the triangle inequality: as Hatzivassiloglou and McKeown (1993) observe, dissimilarity is not transitive.

What motivates the use of the KL divergence, if it is not a true distance metric? We appeal to statistics, information theory, and the maximum entropy principle.

The statistician Kullback (1959) derives the KL divergence from a Bayesian perspective. Let Y be a random variable taking values in $\mathcal{Y}$. Suppose we are considering exactly two hypotheses about Y: H_q is the hypothesis that Y is distributed according to q, and H_r is the hypothesis that Y is distributed according to r. Using Bayes' rule, we can write the posterior probabilities of the two hypotheses as

$$P(H_q|y) = \frac{P(H_q)q(y)}{P(H_q)q(y) + P(H_r)r(y)},$$

and

$$P(H_r|y) = \frac{P(H_r)r(y)}{P(H_q)q(y) + P(H_r)r(y)}.$$

Taking logs of both equations and subtracting, we obtain

$$\log \frac{q(y)}{r(y)} = \log \frac{P(H_q|y)}{P(H_r|y)} - \log \frac{P(H_q)}{P(H_r)}.$$

We can therefore consider $\log (q(y)/r(y))$ to be the information y supplies for choosing H_q over H_r: it is the difference between the logarithms of the posterior odds ratio and the prior odds ratio. $D(q||r)$ is then the average information for choosing H_q over H_r. Thus, the KL divergence does indeed measure the dissimilarity between two distributions, since the greater their divergence is, the easier it is, on average, to distinguish between them.

Another statistical rationale for using the KL divergence is given by Cover and Thomas (1991). Let the *empirical frequency distribution* of a sample $\mathbf{y}$ of length n be the probability mass function $p_{\mathbf{y}}$, where $p_{\mathbf{y}}(y)$ is simply the number of times y showed up in the sample divided by n.

Theorem 2.2 *Let r be a hypothesized source distribution. The probability according to r of observing a sample of length n with empirical frequency distribution q is approximately $b^{-nD(q||r)}$, where b is the base of the logarithm function.*

Therefore, we see that if we are trying to decide between hypotheses $r_1, r_2, \ldots, r_k$ when q is the empirical frequency distribution of the observed sample, then $D(q||r_i)$ gives the relative weight of evidence in favor of hypothesis r_i.

The KL divergence arises in information theory as a measure of coding inefficiency. If Y is distributed according to q, then the average codeword length of the best code for Y is the *entropy* $H(q)$ of q:

$$H(q) = -\sum_{y \in \mathcal{Y}} q(y) \log q(y).$$

However, if distribution r were (mistakenly) used to encode Y, then the average codeword length of the resulting code would increase by $D(q||r)$. Therefore, if the divergence between q and r is large, then q and r must be dissimilar, since it is inefficient (on average) to use r in place of q.

Finally, we look at the maximum entropy argument. The entropy of a distribution can be considered a measure of its uncertainty; distributions for which many outcomes are likely (so that one is "uncertain" which outcome will occur) can only be described by relatively complicated codes. The *maximum entropy principle*, first stated by Jaynes (1957), is to assume that the distribution underlying some observed data is the distribution with the highest entropy among all those consistent with the data – that is, one should pick the distribution that makes the fewest assumptions necessary. If one accepts the maximum entropy principle, then one can use it to motivate the use of the KL divergence in the following manner. The distribution $\tilde{r}(y) = 1/|\mathcal{Y}|$ is certainly the a priori maximum entropy distribution. We can write

$$\begin{aligned} D(q||\tilde{r}) &= \sum_{y\in\mathcal{Y}} q(y)\log q(y) - \sum_{y\in\mathcal{Y}} q(y)\log \tilde{r}(y) \\ &= -H(q) - \log\frac{1}{|\mathcal{Y}|} \\ &= \log|\mathcal{Y}| - H(q). \end{aligned}$$

Maximizing entropy is therefore equivalent to minimizing the KL divergence to the prior $\tilde{r}$ given above, subject to the constraint that one must choose a distribution that fits the data.

To summarize, we have described three motivations for using the KL divergence. For the sake of broad acceptability, we have given both Bayesian arguments (those that refer to priors) and non-Bayesian ones.[2] These are by no means the only reasons. For further background, see Cover and Thomas (1991) and Kullback (1959) for general information, Aczél and Daróczy (1975) for an axiomatic development, and Rényi (1970) for a description of information theory that uses the KL divergence as a starting point.

Some authors (Brown et al., 1992; Church and Hanks, 1990; Dagan, Marcus, and Markovitch, 1995; Luk, 1995) use the *mutual information*, which is the KL divergence between the joint distribution of two random variables and their product distributions. Let A and B be two random variables with probability mass functions $f(A)$ and $g(B)$, respectively, and let $h(A, B)$ be their joint distribution function. Then

$$I(A,B) = D(h||f\cdot g) = \sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} h(a,b)\log\frac{h(a,b)}{f(a)\cdot g(b)}, \tag{2.6}$$

where $\mathcal{A}$ and $\mathcal{B}$ denote the sets of possible values for A and B, respectively. The mutual information measures the dependence of A and B, for if A and B are independent, then $h = f\cdot g$, which implies that the KL divergence between h and $f\cdot g$ is zero by the information inequality (theorem 2.1). We will not give the mutual information further consideration because we do not wish to attempt to estimate joint distributions. Indeed, Church and Hanks (1990) consider two words to be associated if the words occur near each other in some sample of text; but Hatzivassiloglou and McKeown (1993) note that the occurrence of two adjectives in the same noun phrase means that the adjectives cannot be similar. Thus, the information that joint distributions carry about similarity varies too widely across different applications for it to be a generally useful notion for us.

While there are many theoretical reasons justifying the use of the KL divergence, there is a problem with employing it in practice. Recall that for distributions q and r, $D(q||r)$ is infinite if there is some $y' \in \mathcal{Y}$ such that $r(y') = 0$ but $q(y')$ is nonzero. If we know q and r exactly, then this is sensible, since the value y' allows us to distinguish between q and r with absolute confidence. However, often it is the case that we only have estimates $\hat{q}$ and $\hat{r}$ for q and r. If we are not careful with our estimates, then we may erroneously set $\hat{r}(y)$ to zero for some y for which $q(y) > 0$, with the effect that $D(\hat{q}||\hat{r})$ can be infinite when $D(q||r)$ is not.

There are several ways around this problem. One is to use smoothed estimates, as

[2]The often heated debates between Bayesians and non-Bayesians are well known. For example, Skilling (1991, pg. 24) writes, "there is a valid defence [sic] of using non-Bayesian methods, namely incompetence."

described above in section 2.2, for q and r; this is the approach taken in chapter 5. Another is to only calculate the KL divergence between distributions and average distributions. The work described in chapter 3 computes divergences to cluster centroids, which are created by averaging a whole class of objects. Chapter 4 describes experiments where we calculate the total divergence of q and r to their average; we examine some properties of the total divergence in the next subsection.

2.3.2 Total Divergence to the Mean

Equation (2.7) gives the definition of the *total (KL) divergence to the mean*, which appears in Dagan, Lee, and Pereira (1997) (A stands for "average"):

$$A(q,r) = D(q||\frac{q+r}{2}) + D(r||\frac{q+r}{2}), \tag{2.7}$$

where $((q+r)/2)(y) = (q(y)+r(y))/2$. If q and r are two empirical frequency distributions (defined just above theorem 2.2), then $A(q,r)$ can be used as a test statistic for the hypothesis that q and r are drawn from the same distribution.

Using theorem 2.1, we see that $A(q,r) \geq 0$, with equality if and only if $q = r$. $A(q,r)$ is clearly a symmetric function, but does not obey the triangle inequality, as will be shown below.

We can write $A(q,r)$ in a more convenient form by observing that

$$\begin{aligned} D(q\|\frac{q+r}{2}) &= \sum_{y\in\mathcal{Y}} q(y)\log\frac{2q(y)}{q(y)+r(y)} \\ &= \log 2 + \sum_{y\in\mathcal{Y}} q(y)\log\frac{q(y)}{q(y)+r(y)}. \end{aligned}$$

The sum over $y \in \mathcal{Y}$ may be broken up into two parts, a sum over those y such that both $q(y)$ and $r(y)$ are greater than zero, and a sum over those y such that $q(y)$ is greater than zero but $r(y) = 0$. We call these sets *Both* and *Justq*, respectively: $\mathit{Both} = \{y : q(y) > 0, r(y) > 0\}$ and $\mathit{Justq} = \{y : q(y) > 0, r(y) = 0\}$. Then,

$$\begin{aligned} D(q\|\frac{q+r}{2}) &= \log 2 + \sum_{y\in \mathit{Both}} q(y)\log\frac{q(y)}{q(y)+r(y)} + \sum_{y\in \mathit{Justq}} q(y)\log\frac{q(y)}{q(y)+r(y)} \\ &= \log 2 + \sum_{y\in \mathit{Both}} q(y)\log\frac{q(y)}{q(y)+r(y)} + \sum_{y\in \mathit{Justq}} q(y)\log\frac{q(y)}{q(y)} \\ &= \log 2 + \sum_{y\in \mathit{Both}} q(y)\log\frac{q(y)}{q(y)+r(y)}. \end{aligned}$$

A similar decomposition of $D(r\|\frac{q+r}{2})$ into two sums over *Both* and $\mathit{Justr} = \{y : r(y) > 0, q(y) = 0\}$ holds. Therefore, we can write

$$A(q,r) = 2\log 2 + \sum_{y\in \mathit{Both}} \left\{ q(y)\log\frac{q(y)}{q(y)+r(y)} + r(y)\log\frac{r(y)}{q(y)+r(y)} \right\}. \tag{2.8}$$

Equation (2.8) is computationally convenient, for it involves sums only over elements of

Both, as opposed to over all the elements in $\mathcal{Y}$. We will typically consider situations in which *Both* is (estimated to be) much smaller than $\mathcal{Y}$.

Since the two ratios in (2.8) are both less than one, the sum over elements in *Both* is always negative. $A(q,r)$ therefore reaches its maximum when the set *Both* is empty, in which case $A(q,r) = 2\log 2$. This observation makes it easy to see that $A(q,r)$ does not obey the triangle inequality. Let $\mathcal{Y} = \{y_1, y_2\}$. Consider distributions $\tilde{q}$, $\tilde{r}$, and $\tilde{s}$, where

$$\tilde{q}(y_1) = 1,\ \tilde{q}(y_2) = 0; \qquad \tilde{r}(y_1) = \tilde{r}(y_2) = \frac{1}{2}; \qquad \tilde{s}(y_1) = 0,\ \tilde{s}(y_2) = 1.$$

Then $A(\tilde{q},\tilde{r}) + A(\tilde{r},\tilde{s}) = \log 2 + \log(2/3) + 2\log(4/3) = \log 2 + \log(32/27) < 2\log 2$, whereas $A(\tilde{q},\tilde{s}) = 2\log 2$, since the supports for $\tilde{q}$ and $\tilde{s}$ are disjoint. Therefore, $A(\tilde{q},\tilde{r}) + A(\tilde{r},\tilde{s}) \not\geq A(\tilde{q},\tilde{s})$, violating the triangle inequality.

2.3.3 Geometric Distances

If we think of probability mass functions as vectors, so that distribution p is associated with the vector $(p(y_1), p(y_2), \ldots, p(y_N))$ in $\Re^N$, then we can measure the distance between distributions by various geometrically-motivated functions, including the L_1 and L_2 norms and the cosine function. All three of these functions appear quite commonly in the clustering literature (Kaufman and Rousseeuw, 1990; Cutting et al., 1992; Schütze, 1993). The first two functions are true metrics, as the name "norm" suggests.

The L_1 *norm* (also called the "Manhattan" or "taxi-cab" distance) is defined as

$$L_1(q,r) = \sum_{y \in \mathcal{Y}} |q(y) - r(y)|. \tag{2.9}$$

Clearly, $L_1(q,r) = 0$ if and only if $q(y) = r(y)$ for all y. Interestingly, $L_1(q,r)$ bears the following relation, discovered independently by Csiszár and Kemperman, to $D(q||r)$:

$$L_1(q,r) \leq \sqrt{D(q||r) \cdot 2\ln b}\,, \tag{2.10}$$

where b is the base of the logarithm function. Consequently, convergence in KL divergence implies convergence in the L_1 norm. However, we can find a much tighter bound, as follows. By dividing up the sum in equation (2.9) into sums over *Both*, *Justq*, and *Justr* as defined in section 2.3.2, we obtain

$$L_1(q,r) = \sum_{y \in Justq} q(y) + \sum_{y \in Justr} r(y) + \sum_{y \in Both} |q(y) - r(y)|.$$

Since

$$\sum_{y \in Justq} q(y) = 1 - \sum_{y \in Both} q(y) \qquad \text{and} \qquad \sum_{y \in Justr} r(y) = 1 - \sum_{y \in Both} r(y),$$

we can express $L_1(q,r)$ in a form depending only on the elements of *Both*:

$$L_1(q,r) = 2 + \sum_{y \in Both} \left(|q(y) - r(y)| - q(y) - r(y)\right). \tag{2.11}$$

Applying the triangle inequality to (2.11), we see that $L_1(q,r) \leq 2$, with equality if and only if the set *Both* is empty. Also, (2.11) is a convenient expression from a computational point of view, since we do not need to sum over all the elements of $\mathcal{Y}$. We describe experiments using L_1 as distance function in chapter 4.

The L_2 *norm* is the Euclidean distance between vectors. Let $||\cdot||$ denote the usual norm function, $||q(y)|| = \sqrt{\sum_y q(y)^2}$. Then,

$$L_2(q,r) = ||q(y) - r(y)|| = \left(\sum_{y \in \mathcal{Y}} (q(y) - r(y))^2 \right)^{\frac{1}{2}}.$$

Since the L_1 norm bounds the L_2 norm, the inequality of equation (2.10) also applies to the L_2 norm.

Although the L_2 norm appears quite often in the literature, Kaufman and Rousseeuw (1990) write that

> In many branches of univariate and multivariate statistics it has been known for a long time that methods based on the minimization of sums (or averages) of dissimilarities or absolute residuals (the so-called L_1 methods) are much more robust than methods based on sums of squares (which are called L_2 methods). The computational simplicity of many of the latter methods does not make up for the fact that they are extremely sensitive to the effect of one or more outliers. (pg. 117)

We therefore will not give further consideration to the L_2 norm in this thesis.

Finally, we turn to the *cosine function*. This symmetric function is related to the angle between two vectors; the "closer" two vectors are, the smaller the angle between them.

$$\cos(q,r) = \frac{\sum_{y \in \mathcal{Y}} q(y)r(y)}{||q||||r||} \tag{2.12}$$

Notice that the cosine is an inverse distance function, in that it achieves its maximum of 1 when $q(y) = r(y)$ for all y, and is zero when the supports of q and r are disjoint. For all the other functions described above, it is just the opposite: they are zero if and only if $q(y) = r(y)$ for all y, and are greater than zero otherwise. Further analysis of geometric properties of the cosine function and other geometric similarity functions used in information retrieval can be found in Jones and Furnas (1987).

The cosine function is not as efficient to compute as the other functions we have discussed. While the numerator in (2.12) requires only summing over elements of *Both*, the elements of *Justq* and *Justr* must be taken into account in calculating the denominator. It may be desirable to calculate the norms of all distributions as a preprocessing step (we cannot just normalize the vectors because we would violate the constraint that attribute vector components sum to one).

2.3.4 Similarity Statistics

There are many correlation statistics for measuring the association between random variables (Anderberg, 1973, Chapter 4.2). The most well-known of these is the Pearson correlation coefficient; some non-parametric measures are the gamma statistic, Spearman's correlation coefficient, and Kendall's τ coefficient (Gibbons, 1993). The Spearman statistic was used by Finch and Chater (1992) to find syntactic categories, and Kendall's statistic appears in work by Hatzivassiloglou and McKeown (1993) (henceforth H&M) on clustering adjectives. We concentrate on the latter statistic since we will discuss H&M's work in some detail in the next chapter.

Kendall's τ coefficient is based on pairwise comparisons. For every pair of contexts (y_i, y_j), we consider the quantities $\alpha_q^{ij} = q(y_i) - q(y_j)$ and $\alpha_r^{ij} = r(y_i) - r(y_j)$. The pair is a *concordance* if both α_q^{ij} and α_r^{ij} have the same sign, and a *discordance* if their signs differ (if either of these quantities is zero, then the pair is a tie, which is neither a concordance nor a discordance). $\tau(q, r)$ is the difference between the probability of observing a concordance and the probability of observing a discordance, and so ranges between -1 and 1. A value of 1 corresponds to perfect concordance (but not necessarily equality) between q and r, -1 corresponds to perfect discordance, and 0 to no correlation. An unbiased estimator of $\tau(q, r)$ is

$$\hat{\tau}(q, r) = \frac{\text{number of observed concordances} - \text{number of observed discordances}}{\binom{|\mathcal{Y}|}{2}}.$$

In terms of computational efficiency, $\tau(q, r)$ is slightly more expensive then the total divergence to the mean or the L_1 norm. In order to calculate the number of discordances, H&M first order the y's in $\mathcal{Y}$ by their probabilities as assigned by q. Then, they rerank the y's according to the probabilities assigned by r. The number of discordances is then exactly the number of discrepancies between the two orderings. Since we need to sort the set $\mathcal{Y}$ and calculate the number of discrepancies between the two orderings, we spend $O(|\mathcal{Y}| \log_2 |\mathcal{Y}|)$ time to calculate the similarity between q and r. An optimization not noted by H&M is that for all $y' \in \mathit{Both} \cup \mathit{Justq} \cup \mathit{Justr}$ and $y'' \notin \mathit{Both} \cup \mathit{Justq} \cup \mathit{Justr}$ (that is, $q(y'') = r(y'') = 0$), the pair (y', y'') cannot be a discordance – it is a concordance if $y' \in \mathit{Both}$ and a tie otherwise. Therefore, we actually only need to sort $\mathcal{Y}' = \mathit{Both} \cup \mathit{Justq} \cup \mathit{Justr}$, a $O(|\mathcal{Y}'| \log_2 |\mathcal{Y}'|)$ operation. In the case of sparse data, this would be a significant time savings, although we would still be using more than linear time.

2.3.5 An Example

To aid in visualizing the behavior of the salient functions described above, we consider a two-dimensional example where $\mathcal{Y} = \{y_1, y_2\}$. In this situation, $q(y_2) = 1 - q(y_1)$ for any distribution q, so we only need to know the value of a distribution at y_1. In figure 2.1, we have plotted the values of various distance functions with respect to a fixed distribution $r = (.5, .5)$. The horizontal axis represents the probability of y_1, so that .75 on the horizontal axis means the distribution $q = (.75, .25)$. The fixed distribution r is at .5 on the horizontal axis.

As observed above, the KL divergences, the total divergence to the mean, and the L_1 norm are all zero at r and increase as one travels away from r. The cosine function, on the other hand, is 1 at r and decreases as one travels away from r.

Figure 2.1 demonstrates that the KL divergence is not symmetric, for the curve $D(r||q)$ lies above the curve $D(q||r)$. In general, the KL divergence from a sharp to a flat distribution is less than the divergence from a flat to a sharp distribution – a sharp distribution (such as $(.9, .1)$) is one with relatively high values for some of the attributes, whereas a flat distribution resembles the uniform distribution. The intuition behind this behavior is as

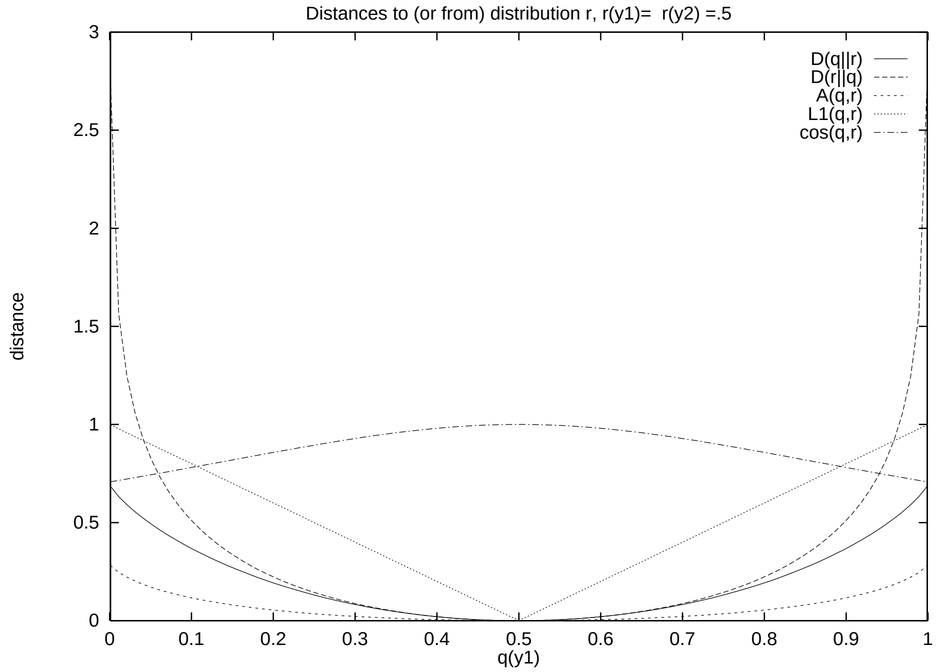

Figure 2.1: Comparison of distance functions

follows. If we assume that the source distribution (the second argument to $D(\cdot||\cdot)$) is flat, then it would be somewhat odd to observe a sharp sample distribution. However, it would be even more surprising to observe a flat sample if we believe that the source distribution is sharp. For instance, suppose the source distribution were (.5,.5). Then, the probability of observing 9 y_1's and 1 y_2 in a sample of length 10 (i.e., a sharp empirical distribution) would be

$$\binom{10}{9}(.5)^9(.5)^1 \approx .01.$$

However, if the source distribution were (.9,.1), then the probability of observing 5 y_1's and 5 y_2's (i.e., a flat empirical distribution) would be

$$\binom{10}{5}(.9)^5(.1)^5 \approx .001.$$

An interesting feature to note is that the curve for $A(q,r)$, the total divergence to the mean, is lower than the KL divergence curves, and that these, in turn, are for the most part lower than the L_1 curve. We speculate that the flatness of $D(q||r)$ and $A(q,r)$ relative to $L_1(q,r)$ around the point $q = r$ indicates that these two functions are somewhat more robust to sampling error, for using $q = r + \epsilon$ (for small ϵ) instead of $q = r$ results in a much greater change in the value of the L_1 norm than in the value of the KL divergence or the total divergence to the mean.

2.4 Summary and Preview

We have now established the groundwork for the results of this thesis. We have explained why we want to use distributions to represent objects, and have described ways to estimate these distributions and to measure the similarity between distributions.

We have been working with conditional probabilities induced by objects over contexts. As mentioned above, "objects" and "contexts" are fairly general notions; for instance, an object might be a document and the contexts might be the set of words that can occur in a document. We will confine our attention to modeling pairs of words, so that $\mathcal{X}$ and $\mathcal{Y}$ are sets of words. In chapters 3 and 4, $\mathcal{X}$ is a set of nouns and $\mathcal{Y}$ is a set of transitive verbs; $C(x, y)$ indicates the number of times x was the direct object of verb y. Chapter 5 considers the bigram case, where $\mathcal{X}$ is the set of all possible words, $\mathcal{Y} = \mathcal{X}$, and $C(x, y)$ denotes the number of times word x occurred immediately before the word y.

Chapter 3

Distributional Clustering

This chapter describes the first of our similarity-based methods for estimating probabilities. The probabilistic, hierarchical distributional clustering scheme detailed here is a model-based approach, where the behavior of objects is modeled by class behavior. The following two chapters describe a nearest-neighbor approach, where we base our estimate of an object's behavior on the behavior of objects most similar to it, so that no class construction is involved.

3.1 Introduction

Much attention has been devoted to the study of clustering techniques, and indeed whole books have been written on the subject (Anderberg, 1973; Hartigan, 1975; Kaufman and Rousseeuw, 1990). Traditional applications of clustering include discovering structure in data and providing summaries of data. We propose to use clustering as a solution to sparse data problems: by grouping data into similarity classes, we create new, generalized sources of information which may be consulted when information about more specific events is lacking. That is, if we wish to estimate the probability of an event $\mathcal{E}$ that occurs very rarely in some sample, then we can base our estimate on the average behavior of the events in $\mathcal{E}$'s class(es); since a class encompasses several data points, estimates of class probability are based on more data than estimates of the probability of a single event. For example, suppose we wish to estimate the graduation rate of Asian-American females enrolled at Westlake High School in Westlake, Ohio. If there is only one Asian-American female at WHS, then we will not have enough data to infer the right rate (we would probably have to guess either 100% or 0%). Suppose, however, that we consider a group of high schools that are similar to WHS (e.g., public high schools in suburban areas in Ohio). Then, we can average together information about Asian-American females attending schools in that group to make a better estimate.

To our knowledge, all clustering algorithms in the natural language processing literature create "hard" or Boolean classes, with every data point belonging to one and only one class. In other words, these algorithms build partitions of the data space. The combinatorial demands of such hard clustering schemes are enormous, as there are $\left\{ {n \atop k} \right\}$ ways to group n observations into k non-empty sets, where

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} i^n$$

is a Stirling number of the second kind (Knuth, 1973). There are a huge number of possible groupings even for small values of k and n: Hatzivassiloglou and McKeown (1993) observe that one can divide twenty-one points into nine sets in approximately 1.23×10^{14} ways. As it turns out, the problem of finding a partition that minimizes some optimization function is NP-complete (Brucker, 1978), so, not surprisingly, most hard clustering algorithms resort to greedy or hill-climbing search to find a good partition.

Greedy and hill-climbing approaches all first create an initial clustering and then iteratively make local changes to the clustering in order to improve the value of some optimization function. Let k be the desired number of clusters. *Update* methods begin with k initial classes chosen in some fashion, and repeatedly move data points from one class to another. The number of clusters therefore stays (about) the same from one iteration to the next. Two special cases of update methods are *medoid* and *centroid* methods, both of which represent clusters by data points. Medoids are actual data points, whereas centroids are "imaginary" data points created by averaging together object distributions (Kaufman and Rousseeuw, 1990). Cluster membership is decided by assigning each object to the closest cluster representative, where "closeness" is measured by some distance function. Each iteration step consists of first moving some representative in order to improve the value of the optimization criterion, and then updating cluster memberships.

Non-update methods, where the number of clusters varies during the course of the clustering, include *divisive* and *agglomerative* clustering. Divisive algorithms start with one universal class to which all the data points belong; each iteration involves choosing one of the current set of classes to split into two new classes. Agglomerative algorithms, in contrast, begin with each data point belonging to its own class; then, in each iteration step, some pair of current classes is merged to form a new, larger class. In either case, the choice of which class to split or which classes to merge is generally made by picking the class or classes whose division or combination results in the largest improvement in the optimization function, and the process stops once k clusters have been formed.

Both divisive algorithms and agglomerative algorithms, if allowed to run until all classes have been merged into one, readily yield *hierarchical* clusterings, which can be represented by *dendrograms* (essentially, binary trees). At the root of the dendrogram is the class containing all the data points (the first class considered in the divisive case and the last class formed in the agglomerative case). Each node η in the dendrogram represents a class, denoted by class(η). Nodes η_1 and η_2 are children of node η' if at some iteration step either class(η') was divided into class(η_1) and class(η_2), or class(η_1) and class(η_2) were agglomerated into class(η'), depending on which type of clustering algorithm was used.

While the class hierarchy produced may of course itself be of interest, an appealing aspect of hierarchical clustering is that it provides an attractive solution to the problem of deciding on the right number of clusters. The partitioning methods mentioned above generally take the number of clusters k as an input parameter rather than deciding what the right number of clusters is. As Anderberg (1973) writes, "Hierarchical clustering methods give a configuration for every number of clusters from one (the entire data set) up to the number of entities (each cluster has only one member)" (pg. 15). However, both Anderberg (1973) and Kaufman and Rousseeuw (1990) express reservations about hierarchical methods:

> A hierarchical method suffers from the defect that it can never repair what was done in previous steps. Indeed, once an agglomerative algorithm has joined two objects, they cannot be separated....Also, whatever a divisive algorithm has split up cannot be reunited. The rigidity of hierarchical methods is both the

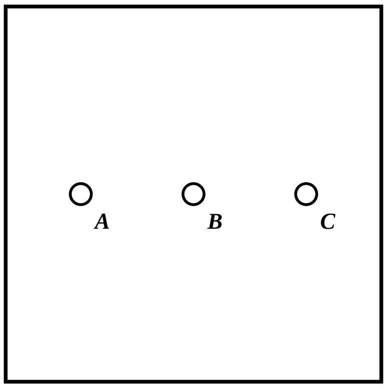

Figure 3.1: An ambiguous case

> key to their success (because it leads to small computation times) and their main disadvantage (the inability to correct erroneous decisions)." (Kaufman and Rousseeuw, 1990, pp. 44-45)

We propose a novel "soft" (probabilistic) hierarchical clustering method that overcomes this rigidity problem. Instead of each data point belonging to one and only one class, we assign probabilities of class membership, with every data point belonging to every class with positive probability. Since we reestimate membership probabilities at each iteration, there is no sense in which data points can be permanently assigned to the same or separate classes.

Probabilistic clusterings have another advantage: they provide a more descriptive summary of the data. Consider the situation depicted in figure 3.1, where circle B is halfway between A and C. Suppose that two clusters are desired. A hard clustering is forced to associate B with only one of the other circles, say, A. It then reports that the partition found is $\{\{A, B\}, \{C\}\}$, which does not convey the information that B could just as well have been grouped with C. A soft clustering, on the other hand, can state that B belongs to A's cluster and C's cluster with equal probability, and so can express the ambiguity of the situation.

In brief, our clustering method is a centroid-based, probabilistic, divisive, hierarchical algorithm for associating abstract objects by learning their distributions. Each class is represented by a centroid, which is placed at the cluster's (weighted) center of mass. For each object x and each centroid c, we calculate a membership probability $P(c|x)$ that x belongs to c. Our method begins by creating a single centroid, with each object belonging to that centroid with probability one, and then iteratively splits one of the current centroids and reestimates membership probabilities. The creation of child centroids from parent centroids creates a hierarchy of classes in the obvious way.

As decided upon in section 2.1, objects (both data points and centroids) will be represented by distributions over a set $\mathcal{Y}$ of contexts. We will use the KL divergence, discussed at length in section 2.3.1, as distance function. Our optimization function is the *free energy*, a quantity motivated by statistical physics; the algorithm uses deterministic annealing to find phase transitions of the free energy, and splits cluster centroids at these transitions. Each time we update the annealing parameter, we reestimate the location of the cluster centroids and the membership probabilities for each object.

We shall be especially interested in the problem of clustering words, although our theoretical results will be described in a general fashion. We re-emphasize that our clustering method can be used for clustering any objects that can be described as distributions, and indeed future work involves employing our techniques for clustering documents. We evaluate our method on tasks involving the prediction of object-verb pairs, and find that it greatly reduces error rate, especially in cases where traditional methods such as Katz's back-off method (see section 2.2) would fail.

3.2 Word Clustering

Methods for automatically classifying words according to their contexts are of both scientific and practical interest. The scientific questions arise in connection with distributional views of linguistic (particularly lexical) structure and also in relation to the question of lexical acquisition. From a practical point of view, word classification addresses issues of data sparsity and generalization in statistical language models, especially models used to decide among alternative analyses proposed by a grammar.

It is well known that a simple tabulation of frequencies of certain words participating in certain configurations (for example, frequencies of pairs of transitive main verbs and head nouns of the verbs' direct objects), cannot be reliably used for comparing the likelihoods of different alternative configurations. The problem is that for large samples, the number of possible joint events is much larger than the number of event occurrences in the sample, so many events occur rarely or even not at all. Frequency counts thus yield unreliable estimates of their probabilities.

Hindle (1990) proposed dealing with the data sparseness problem by estimating the likelihood of unseen events from that of "similar" events that have been seen. For instance, one may estimate the likelihood of a particular adjective modifying a noun from the likelihoods of that adjective modifying similar nouns. This requires a reasonable definition of noun similarity and a method for incorporating the similarity into a probability estimate. In Hindle's proposal, words are similar if there is strong statistical evidence that they tend to participate in the same events. His notion of similarity seems to agree with our intuitions in many cases, but it is not clear how to use this notion to construct word classes and corresponding models of association.

In this chapter, we build a similarity-based probability model out of two parts: a model of the association between words and certain hidden classes, and a model of the behavior of these classes. Some researchers have built such models from preexisting sense classes constructed by humans; for example, Resnik (1992) uses WordNet, and Yarowsky (1992b) works with Roget's thesaurus. As mentioned in chapter 2, however, we are interested in ways to derive classes directly from distributional data. Resnik's thesis contains a discussion of the relative advantages of the two approaches (Resnik, 1993).

In what follows, we will consider two sets of words, the set $\mathcal{X}$ of nouns, and the set $\mathcal{Y}$ of transitive verbs. We are interested in the object-verb relation: the pair (x, y) denotes the event that noun x occurred as the head noun of the direct object of verb y. Our raw knowledge about the relation consists of the frequencies $C(x, y)$ of particular pairs (x, y) in the required configuration in a training corpus. Some form of text analysis is required to collect these pairs. The counts used in our first experiment were derived from newswire text automatically parsed by Hindle's parser Fidditch (Hindle, 1994). Later, we constructed similar frequency tables with the help of a statistical part-of-speech tagger (Church, 1988) and tools for regular expression pattern-matching on tagged corpora (Yarowsky, 1992a). We have not compared the accuracy and coverage of the two methods or studied what biases they introduce, although we took care to filter out certain systematic errors (for instance, subjects of complement clauses for report verbs like "say" were incorrectly parsed as direct objects).

We only consider the problem of classifying nouns according to their distribution as direct objects of verbs; the converse problem is formally similar. For the noun classification problem, the empirical distribution of a noun x is given by the conditional density

$$P_{MLE}(y|x) = \frac{C(x,y)}{\sum_y C(x,y)} = \frac{C(x,y)}{C(x)},$$

where $C(z)$ denotes the number of times event z occurred in the training corpus. The problem we study is how to use the $P_{MLE}(\cdot|x)$ to classify the $x \in \mathcal{X}$. Our classification method will construct a set $\mathcal{C}$ of clusters c and cluster membership probabilities $\tilde{P}(c|x)$. Each cluster c is associated with a cluster centroid distribution $\tilde{P}(y|c)$, which is a discrete density over $\mathcal{Y}$ obtained by computing a weighted average of the noun distributions $P_{MLE}(\cdot|x)$. We will move freely between describing a noun (or centroid) as x (or c) and as $P_{MLE}(\cdot|x)$ (or $\tilde{P}(\cdot|c)$).

To cluster nouns x according to their conditional verb distributions $P_{MLE}(\cdot|x)$, we need a measure of similarity between distributions. We use for this purpose the KL divergence from section 2.3.1:

$$D(q||r) = \sum_{y \in \mathcal{Y}} q(y) \log \frac{q(y)}{r(y)}.$$

The KL divergence is a natural choice for a variety of reasons, most of which we have already discussed in section 2.3.1. As mentioned there, $D(q||r)$ measures how inefficient on average it would be to use a code based on r to encode a variable distributed according to q. With respect to our problem, $D(P_{MLE}(\cdot|x)||\tilde{P}(\cdot|c))$ thus gives us the loss of information in using the centroid distribution $\tilde{P}(\cdot|c)$ instead of the empirical distribution $P_{MLE}(\cdot|x)$ when modeling noun x. Furthermore, minimizing the KL divergence yields cluster centroids that are a simple weighted average of member distributions, as we shall see.

One technical difficulty is that $D(q||r)$ is infinite when $r(y) = 0$ but $q(y) > 0$. Due to sparse data problems, it is often the case that $P_{MLE}(y|x)$ is zero for a particular pair (x, y). We could sidestep this problem by smoothing zero frequencies, perhaps using one of the methods described in section 2.2. However, this is not very satisfactory because one of the goals of our work is precisely to avoid data sparsity problems by grouping words into classes. As it turns out, the difficulty is avoided by our clustering technique: instead of computing the KL divergence between individual word distributions, we only calculate divergences between word distributions and cluster centroids. Since centroids are average distributions, they are guaranteed to be nonzero whenever the word distributions are. This is a useful advantage of our method over techniques that need to compare pairs of individual objects, since estimates for individual objects are prone to inaccuracies due to data sparseness.

The organization of the rest of this chapter is as follows. We develop the theoretical basis for our clustering algorithm in section 3.3. We present some example clusterings in section 3.4 in order to get a sense of the qualitative performance of our algorithm. Section 3.5 presents two evaluations of the ability of our cluster-based probability estimation method to estimate word pair probabilities, especially in situations where data is sparse; we show that indeed, our method does a good job of modeling. Finally, in section 3.6 we review other work in the NLP community on clustering words, and briefly touch upon soft clustering methods from other fields.

3.3 Theoretical Basis

Our general problem can be seen as that of learning the joint distribution $P(x,y)$ of pairs in $\mathcal{X} \times \mathcal{Y}$ from a large sample. The training data is a sample S of n independently drawn pairs

$$S_i = (x_{j_i}, y_{l_i}), \qquad 1 \leq i \leq n.$$

We assume that each $x_j \in \mathcal{X}$ and $y_l \in \mathcal{Y}$ occurs in the sample at least once (we cannot train a model for x or y if we have no information about them).

The line of argument in this section proceeds as follows. We first set up the general form of our cluster-based probability model. We determine the two principles, minimum distortion and maximum entropy, that guide our search for the proper parameter settings for the model, and combine these two principles into the *free energy* function. Sections 3.3.1 and 3.3.2 go into the details of how we set the parameters by maximizing entropy and minimizing distortion. Finally, section 3.3.3 describes how searching for phase transitions of the free energy yields a hierarchical clustering.

In order to estimate the likelihood of the sample, we need a probability model $\tilde{P}(x,y) = \tilde{P}(x)\tilde{P}(y|x)$. We would like to find a set of clusters $\mathcal{C}$, each represented by a cluster centroid c, such that each conditional distribution $\tilde{P}(y|x)$ can be decomposed as

$$\tilde{P}(y|x) = \sum_{c \in \mathcal{C}} \tilde{P}(c|x)\tilde{P}(y|c). \tag{3.1}$$

$\tilde{P}(c|x)$ is the *membership probability* that x belongs to c, and $\tilde{P}(y|c)$ is y's probability according to the *centroid distribution* for c: as stated above, centroids are representative objects, and so form a distribution over $\mathcal{Y}$ just like objects do. Ideally, the objects that belong most strongly to a given cluster would be similar to one another.

According to equation (3.1), then, we estimate the probability of y given x by taking an average of the centroid distributions, weighting each $\tilde{P}(y|c)$ by the probability that x belongs to c. We thus make a Markovian assumption that the association of x and y is made solely through the clusters, that is, that y is conditionally independent of x given c. The cluster model drastically reduces the dimension of the model space, since the number of (c,x) and (c,y) pairs should be much lower than the number of possible (x,y) pairs.

Given the decomposition of $\tilde{P}(y|x)$ in equation (3.1), we can write the likelihood assigned by our model to a pair as

$$\tilde{P}(x,y) = \sum_{c \in \mathcal{C}} \tilde{P}(x)\tilde{P}(c|x)\tilde{P}(y|c). \tag{3.2}$$

We will assume that the marginals for $x \in \mathcal{X}$ are not part of our model and so can be considered fixed; to indicate this, we will write $P(x)$ instead of $\tilde{P}(x)$. Without loss of generality, we assume that $P(x)$ is greater than zero for all x. In order to flesh equation (3.2) out, then, we need only find suitable forms for the cluster membership distributions $\tilde{P}(c|x)$ and centroid distributions $\tilde{P}(y|c)$. We will be guided by two principles: first, that our model should fit the data well (otherwise, our model is not useful), and second, that our model should make as few assumptions as possible (otherwise, our model is not general).

Goodness of fit is determined by the *distortion* of the model. Equation (3.1) estimates the probability of y given x by randomly selecting a cluster c according to distribution $\tilde{P}(c|x)$, and then using $\tilde{P}(y|c)$ to estimate the (conditional) probability of y. Recall from section 2.3.1 that $D(P_{MLE}(\cdot|x)||\tilde{P}(\cdot|c))$ measures the inefficiency of using c's distribution rather than x's maximum likelihood distribution to code for x. The distortion $\mathcal{D}$ is the average coding loss incurred by our model:

$$\mathcal{D} = \sum_x P(x) \sum_c \tilde{P}(c|x) d(x,c), \tag{3.3}$$

where $d(x,c)$ is notational shorthand for $D(P_{MLE}(\cdot|x)||\tilde{P}(\cdot|c))$.

As it turns out, the distortion equation does not give us enough information to find good closed-form expressions for the membership probabilities. In fact, without any other constraints, the cluster system that minimizes distortion is the one in which there is one centroid placed on top of each object, with each object belonging only to the centroid it coincides with. Therefore, we add the requirement that the membership assignments make the fewest assumptions possible, that is, that the probability that an object belongs to a centroid should not be any higher than it needs to be. This requirement corresponds to the maximum entropy principle, described in section 2.3.1. Therefore, we wish to maximize the *configuration entropy*

$$H = -\sum_x P(x) \sum_c \tilde{P}(c|x) \log \tilde{P}(c|x), \tag{3.4}$$

which is the average entropy of the membership probabilities.

We can combine distortion and entropy into a single function, the *free energy*, which appears in work on statistical mechanics (Rose, Gurewitz, and Fox, 1990):

$$F = \mathcal{D} - H/\beta. \tag{3.5}$$

This function is not arbitrary; indeed, at maximum entropy points (see section 3.3.1), we can show that

$$H = -\frac{\partial F}{\partial T} \text{ and} \tag{3.6}$$

$$D = \frac{\partial \beta F}{\partial \beta}, \tag{3.7}$$

$$\tag{3.8}$$

where $T = 1/\beta$. The *minima* of F are of special interest to us, since such points represent a balance between the "disordering" force of maximizing entropy and the "ordering" force of minimizing distortion. In fact, in statistical mechanics, the probability of finding a system in a given configuration is a negative exponential in F, so the system is most likely to be found in its minimal free energy configuration. β is a free parameter whose interpretation we will leave for later.

Suppose we fix the number of clusters $|\mathcal{C}|$. Clearly, (local) minima of F occur when the entropy is at a (local) maximum and, simultaneously, the distortion is at a (local) minimum (although critical points of F need not correspond to critical points of $\mathcal{D}$ and H). However, it is difficult to jointly maximize entropy and minimize distortion, since the location of cluster centroids affects the membership probabilities, and vice versa; that is, the $\tilde{P}(y|c)$ and the $\tilde{P}(c|x)$ are not independent. We therefore simplify the search for minima of F

by breaking up the estimation process into two steps. First, we hold the distortion and centroid distributions fixed, and maximize the entropy subject to these constraints. Since the distortion is regarded as constant in this step, maximizing entropy corresponds to a reduction in free energy. Second, we fix the membership probabilities at the values derived in the first step, and thus can treat the entropy as a constant. We then find a critical point of F with respect to the centroid distributions; it turns out that this critical point is in fact a minimum of the distortion, and therefore free energy is reduced once again. Moving the centroid distributions may change the values of the membership probabilities that maximize entropy, though, and so we repeat these two steps until a stable configuration is reached. This two-step estimation iteration is reminiscent of the EM (Estimation-Maximization) algorithm (Dempster, Laird, and Rubin, 1977) commonly used to find maximum likelihood solutions.

Before we continue, we review the notation that will be used in the following sections. Model probabilities are always marked with a tilde ($\tilde{P}$). The model parameters are the membership probabilities $\tilde{P}(c|x)$ and the centroid distributions $\tilde{P}(y|c)$. The object marginal probabilities $\tilde{P}(x) = P(x)$ are not considered part of the model, and so are regarded as positive constants throughout.[1] The centroid marginals $\tilde{P}(c)$ are given by $\tilde{P}(c) = \sum_x \tilde{P}(c|x)P(x)$; this form ensures that $\sum_x \tilde{P}(x|c) = 1$. Empirical frequency distributions are denoted by P_{MLE} and are considered fixed by the data. By assumption, for all y there exists an object x such that $P_{MLE}(y|x) > 0$. The quantity $d(x,c)$ is shorthand for the KL divergence $D(P_{MLE}(\cdot|x)||\tilde{P}(\cdot|c))$. We summarize this information in table 3.1.

Quantity	Value	Notes
$\tilde{P}(c\|x)$	?	(to be determined)
$\tilde{P}(y\|c)$	?	(to be determined)
$\tilde{P}(x)$	$P(x)$	fixed at positive values
$\tilde{P}(c)$	$\sum_x \tilde{P}(c\|x)P(x)$	determined by $\tilde{P}(c\|x)$
$\tilde{P}(x\|c)$	$\tilde{P}(c\|x)P(x)/\tilde{P}(c)$	determined by $\tilde{P}(c\|x)$
$P_{MLE}(y\|x)$	$C(x,y)/C(x)$	fixed by data; $\forall y, \exists x : P_{MLE}(y\|x) > 0$
$d(x,c)$	$D(P_{MLE}(\cdot\|x)\|\|\tilde{P}(\cdot\|c))$	determined by $\tilde{P}(y\|c)$

Table 3.1: Summary of common quantities

We will use natural logarithms in this chapter, so that the base of the logarithm function is e; using another base would not not substantially alter our results, but we would have extra constant factors in most of our expressions. The next two subsections assume that the number of clusters has been fixed.

3.3.1 Maximum-Entropy Cluster Membership

This section addresses the first parameter estimation step of finding the cluster membership probabilities $\tilde{P}(c|x)$ that maximize the configuration entropy, and hence reduce the free energy, assuming that the distortion and the centroid distributions are fixed (it does not

[1] Our implementation sets $P(x) = 1/|\mathcal{X}|$ instead of $P_{MLE}(x)$, since we are interested in distributional modeling without regard to the frequencies of particular nouns.

suffice simply to hold the centroid distributions fixed, since we see from equation (3.3) that the distortion depends on the membership probabilities, too). We will make the further assumption that for all centroids c, $\tilde{P}(y|c) > 0$ for all y; this assumption is justified in the next section.

Recall the definition of the configuration entropy H from equation (3.4):

$$H = -\sum_x P(x) \sum_c \tilde{P}(c|x) \log \tilde{P}(c|x).$$

We wish to maximize this quantity subject to two constraints: the normalization constraint that $\sum_c \tilde{P}(c|x) = 1$ for all x, and the distortion constraint that $\mathcal{D} = K$ for some constant K. We therefore take the variation of the function H^+:

$$H^+ = H - \sum_x \alpha_x \left(\sum_c \tilde{P}(c|x) - 1 \right) - \beta \left(\sum_x P(x) \sum_c \tilde{P}(c|x) d(x,c) - K \right),$$

where α_x and β are Lagrange multipliers. It is important to note that we are using β both here as a multiplier and as a normalization term in the free energy (3.5).

We now calculate the partial derivative of H^+ with respect to a given membership probability $\tilde{P}(c|x)$, since fixing the centroid distributions means that the $\tilde{P}(c|x)$ are independent (except for their association through the fixed distortion).

$$\begin{aligned}
\frac{\partial H^+}{\partial \tilde{P}(c|x)} &= \frac{\partial H}{\partial \tilde{P}(c|x)} - \alpha_x - \beta P(x) d(x,c) \\
&= -\left(P(x) \tilde{P}(c|x) \frac{1}{\tilde{P}(c|x)} + P(x) \log \tilde{P}(c|x) \right) - \alpha_x - \beta P(x) d(x,c) \\
&= -P(x) \left(1 + \log \tilde{P}(c|x) + \frac{\alpha_x}{P(x)} + \beta d(x,c) \right)
\end{aligned}$$

(there is no problem with division by $P(x)$ since we assumed all object marginals are positive).

At critical points of H^+, we have that $\partial H^+ / \partial \tilde{P}(c|x) = 0$. This allows us to solve for $\tilde{P}(c|x)$:

$$\tilde{P}(c|x) = e^{-\beta d(x,c)} e^{-(1+\alpha')},$$

where $\alpha' = \alpha_x / P(x)$. Since α' is meant to insure the normalization of $\tilde{P}(c|x)$, α' must be set to a value such that the following is satisfied:

$$e^{1+\alpha'} = \sum_c e^{-\beta d(x,c)} \stackrel{def}{=} Z_x$$

(Z is standard notation for partition (normalization) functions; the name comes from the German *Zustandsumme*). We therefore have a closed-form solution for the membership probabilities:

$$\tilde{P}(c|x) = \frac{e^{-\beta d(x,c)}}{Z_x}. \tag{3.9}$$

It was shown by Jaynes (1983) that the exponential form (3.9) gives not just a critical point but the maximum of the entropy, and so we have a maximum entropy estimate of membership probability, as desired. The expression (3.9) is intuitively satisfying because it makes the membership probabilities dependent on distance (in the KL divergence sense): the farther x is from c, the less likely it is that x belongs to c. Furthermore, given that the centroid distributions were fixed at positive values for all y, $d(x,c)$ is always defined, which means that all membership probabilities are positive; each object has some degree of association with each cluster. For each $\tilde{P}(c|x)$, we need to calculate $d(x,c)$ which is a sum over all $y \in \mathcal{Y}$, so the time to update all the membership probabilities is $O(|\mathcal{X}||\mathcal{C}||\mathcal{Y}|)$. However, if the object distributions are sparse, then the computation of $d(x,c)$ will be significantly faster.

There is a pleasing relationship between expression (3.9) for $\tilde{P}(c|x)$ and an estimate given by theorem 2.2, restated here:

Theorem 2.2 *Let r be a hypothesized source distribution. The probability according to r of observing a sample of length n with empirical frequency distribution q is approximately $b^{-nD(q||r)}$, where b is the base of the logarithm function.*

Thus, the maximum entropy membership probability $\tilde{P}(c|x) = e^{-\beta d(x,c)}/Z_x$ corresponds to the probability of observing object distribution $P_{MLE}(y|x)$ if the source distribution is assumed to be the centroid $\tilde{P}(y|c)$, except that β has replaced the sample size n. Therefore, if we regard β not as a Lagrange multiplier but as a free parameter, we can in some sense control the sample size ourselves. If we use a high value of β, then we express strong belief in the maximum likelihood estimate P_{MLE} (as would be the case for a very large sample), so that the probability that x belongs to a centroid c is negligible unless $d(x,c)$ is very small. Conversely, a low value of β is equivalent to a small sample, in which case we do not trust the MLE and so allow $\tilde{P}(c|x)$ to be high even if x is relatively distant from c. Section 3.3.3 describes how we vary β in order to derive a hierarchical clustering.

We conclude this section by observing that at the maximum entropy membership probabilities, the free energy can be rewritten as follows:

$$\begin{aligned}
F &= \mathcal{D} - \frac{1}{\beta}H \\
&= \sum_x \sum_c P(x)\tilde{P}(c|x)\left(d(x,c) + \frac{1}{\beta}\log \tilde{P}(c|x)\right) \\
&= \sum_x \sum_c P(x)\tilde{P}(c|x)\left(d(x,c) - d(x,c) - \frac{1}{\beta}\log Z_x\right) \quad \text{(substitution of (3.9))} \\
&= -\frac{1}{\beta}\sum_x P(x)\log Z_x \sum_c \tilde{P}(c|x) \\
&= -\frac{1}{\beta}\sum_x P(x)\log Z_x, \qquad (3.10)
\end{aligned}$$

where the last step is justified since we ensured the normalization of the maximum-entropy membership probabilities. By simple differentiation, it is easy to see that if we set $T = 1/\beta$, then $\partial F/\partial T = -\beta F - \beta\mathcal{D} = -H$, and $\partial(\beta F)/\partial\beta = \mathcal{D}$. This gives us equations (3.6) and (3.7), as desired.

3.3.2 Minimum-Distortion Cluster Centroids

We now proceed with the second estimation step. We fix the membership probabilities $\tilde{P}(c|x)$ at their maximum entropy values, calculated above, so that the configuration entropy can now be considered a constant, and the expression for the free energy is given by equation (3.10).

Now that the membership probabilities have been fixed, the individual centroid distributions are all independent and we just need to find values for them that minimize F, subject to the constraint that $\sum_y \tilde{P}(y|c) = 1$ for all centroids. What we will do is first find a critical point of F (equations (3.11) through (3.14)), and then prove in lemma 3.1 that this critical point is in fact a minimum by showing that it minimizes the distortion $\mathcal{D}$.

In order to find a critical point of F, we take partial derivatives of

$$F^+ = -\frac{1}{\beta}\sum_x P(x)\log Z_x - \sum_c \gamma_c \left(\sum_y \tilde{P}(y|c) - 1\right),$$

where γ_c is yet another Lagrange multiplier and we use expression (3.10) for F.

The partial derivative of F^+ with respect to a given $\tilde{P}(y|c)$ is calculated as follows:

$$\begin{aligned}
\frac{\partial F^+}{\partial \tilde{P}(y|c)} &= -\frac{1}{\beta}\sum_x P(x)\cdot\frac{1}{Z_x}\cdot\frac{\partial}{\partial \tilde{P}(y|c)}\left(\sum_{c'} e^{-\beta d(x,c')}\right) - \gamma_c \\
&= -\frac{1}{\beta}\sum_x P(x)\frac{e^{-\beta d(x,c)}}{Z_x}\left(-\beta\frac{\partial d(x,c)}{\partial \tilde{P}(y|c)}\right) - \gamma_c \\
&= \sum_x P(x)\tilde{P}(c|x)\frac{\partial d(x,c)}{\partial \tilde{P}(y|c)} - \gamma_c. \qquad (3.11)
\end{aligned}$$

The variation of $d(x,c)$ with respect to $\tilde{P}(y|c)$ is

$$\begin{aligned}
\frac{\partial d(x,c)}{\partial \tilde{P}(y|c)} &= \frac{\partial}{\partial \tilde{P}(y|c)}\left(\sum_{y'} P_{MLE}(y'|x)\log\frac{P_{MLE}(y'|x)}{\tilde{P}(y'|c)}\right) \\
&= -\frac{P_{MLE}(y|x)}{\tilde{P}(y|c)}, \qquad (3.12)
\end{aligned}$$

so the centroid distribution term $\tilde{P}(y|c)$ reappears. Substituting (3.12) into (3.11), we have

$$\begin{aligned}
\frac{\partial F^+}{\partial \tilde{P}(y|c)} &= \sum_x P(x)\tilde{P}(c|x)\left(-\frac{P_{MLE}(y|x)}{\tilde{P}(y|c)}\right) - \gamma_c \\
&= -\frac{1}{\tilde{P}(y|c)}\sum_x P(x)\tilde{P}(c|x)P_{MLE}(y|x) - \gamma_c.
\end{aligned}$$

At a critical point of F^+, the partial derivative of F^+ must be 0, which allows us to solve for $\tilde{P}(y|c)$:

$$\tilde{P}(y|c) = \frac{1}{\gamma_c}\sum_x P(x)\tilde{P}(c|x)P_{MLE}(y|x) \qquad (3.13)$$

The multiplier γ_c is meant to enforce the constraint that $\sum_y \tilde{P}(y|c) = 1$, so

$$1 = \sum_y \frac{1}{\gamma_c}\sum_x P(x)\tilde{P}(c|x)P_{MLE}(y|x)$$

$$= \frac{1}{\gamma_c} \sum_x P(x)\tilde{P}(c|x).$$

Therefore, $\gamma_c = \sum_x P(x)\tilde{P}(c|x) = \tilde{P}(c)$; upon substitution of this into (3.13), we finally obtain the centroid distributions:

$$\tilde{P}(y|c) = \sum_x \tilde{P}(x|c) P_{MLE}(y|x). \tag{3.14}$$

We thus have a natural expression for a cluster centroid c: it is an average over all data points x, weighted by the Bayes inverse of the probability that x belongs to c. The Bayes inverses are all positive since the maximum-entropy membership probabilities are, so the centroid distribution cannot be zero for any y since we assume that $P_{MLE}(y|x)$ is nonzero for at least one x. It is clear that the time required to update all the centroid distributions is $O(|\mathcal{Y}||\mathcal{C}||\mathcal{X}|)$ in the worst case; again, however, the computation is much faster if the object distributions are sparse.

Now, expression (3.14) gives us the unique critical point of F when the entropy is held fixed; but is the free energy actually reduced at this point? Our goal, after all, is to look for minima of F. Since the entropy was held fixed, it suffices to show that the centroid distributions (3.14) yield a minimum of the distortion, which we do in the following lemma.

Lemma 3.1 *If the distortion $\mathcal{D}$ has exactly one critical point with respect to centroid distributions $\tilde{P}(y|c)$, $0 \leq \tilde{P}(y|c) \leq 1$, then that critical point is the unique minimum of $\mathcal{D}$, assuming the cluster membership probabilities $\tilde{P}(c|x)$ are fixed.*

Proof. Since

$$\begin{aligned}
\mathcal{D} &= \sum_x P(x) \sum_c \tilde{P}(c|x) d(x,c) \\
&= \sum_x P(x) \sum_c \tilde{P}(c|x) \sum_y P_{MLE}(y|x) \log P_{MLE}(y|x) - \\
&\qquad \sum_x P(x) \sum_c \tilde{P}(c|x) \sum_y P_{MLE}(y|x) \log \tilde{P}(y|c),
\end{aligned}$$

and the centroid distributions are independent when the membership probabilities are fixed, it is sufficient to maximize for each centroid c the quantity

$$\begin{aligned}
\mathcal{D}_c &= \sum_x P(x)\tilde{P}(c|x) \sum_y P_{MLE}(y|x) \log \tilde{P}(y|c) \\
&= \sum_y \left(\log \tilde{P}(y|c)\right) \sum_x P(x)\tilde{P}(c|x) P_{MLE}(y|x) \\
&= \sum_y \log \left(\tilde{P}(y|c)^{Q(c,y)}\right) \\
&= \log \left(\prod_y \tilde{P}(y|c)^{Q(c,y)}\right),
\end{aligned}$$

where $Q(c,y) = \sum_x P(x)\tilde{P}(c|x)P_{MLE}(y|x)$ does not depend on $\tilde{P}(y|c)$. But since the logarithm is a strictly increasing function, we need only find a maximum of the product

$$\prod_y \tilde{P}(y|c)^{Q(c,y)}. \tag{3.15}$$

Observe that this product (unlike the logarithm, which is why we had to do all this equation rewriting) is continuous on the domain $\{\tilde{P}(y|c) : 0 \leq \tilde{P}(y|c) \leq 1\}$, which is closed and bounded. Therefore, we know from analysis that (3.15) achieves both its maximum and its minimum on its domain. Since clearly every point on the boundary of the domain yields a minimum value (zero), the unique critical point must be the maximum of (3.15) and thus the minimum of $\mathcal{D}$. ∎

Now, since the fixed membership probabilities determine the entropy, any critical point of F must also be a critical point of the distortion because $\partial F = \partial \mathcal{D}$ if H is a constant. Therefore, the centroid distributions (3.14) define the unique critical point of the distortion, and application of lemma 3.1 tells us that this is indeed the minimum of $\mathcal{D}$. Thus, we have succeeded in finding centroid distributions which minimize distortion and therefore reduce the free energy.

3.3.3 Hierarchical Clustering

In the previous two sections, we developed maximum entropy estimates for membership probabilities and minimum distortion estimates for centroid distributions:

$$\begin{aligned} \tilde{P}(c|x) &= \exp(-\beta d(x,c))/Z_x && (3.9)\text{, and} \\ \tilde{P}(y|c) &= \sum_x \tilde{P}(x|c) P_{MLE}(y|x) && (3.14). \end{aligned}$$

Our search for minima of F at a fixed β is a two-step iteration described in section 3.3. First, we set the membership probabilities at their maximum entropy values (3.9), using the current centroid distributions. Then, we plug these membership probabilities into (3.14) to update the centroid distributions. We repeat this two-step cycle until the parameters converge to steady states.

Now, this two-step iteration lets us find cluster centroids and membership probabilities for a fixed number of clusters. However, we have not yet shown how the number of clusters is chosen. The inclusion of the parameter β in the free energy expression

$$F = \mathcal{D} - H/\beta$$

suggests the use of a *deterministic annealing* procedure for clustering (Rose, Gurewitz, and Fox, 1990), in which the number of clusters is determined through a sequence of phase transitions by continuously increasing β according to an *annealing schedule.*

As discussed in section 3.3.1, β plays a role similar to sample size and thus controls the importance of the distance function $d(x,c)$. However, it will now be fruitful to think of β as the inverse of temperature. At the high temperature limit (low β), the entropy H has the biggest role in minimizing the free energy, so a system consisting of only one cluster centroid is preferred. At the low temperature limit (high β), the distortion dominates and the minimum-energy configuration is then the one where we have one centroid placed on top of every data point, with each data point belonging with probability one to the centroid it coincides with. Thus, the system has "cooled down" to the point where the freedom of objects to associate with distant centroids has disappeared. Between these two extremes, there must be critical values of β at which *phase transitions* occur; that is, when the natural solution involves including more centroids.

We find these phase transitions by taking a cluster c and a *twin* c^* of c such that the centroid $\tilde{P}(\cdot|c^*)$ is a small random perturbation of $\tilde{P}(\cdot|c)$. Below the critical β at which c splits, the membership and centroid iterative reestimation procedure will make $P(\cdot|c)$ and $P(\cdot|c^*)$ converge, from which we infer that c and c^* are really the same cluster. But if β is above the critical value for c, the two centroids will diverge, giving rise to two children of c.

A sketch of our clustering procedure appears in figure 3.2. We start with very low β and a single cluster whose centroid is the average of all noun distributions (and so is guaranteed to be nonzero for all y). For any given β, we have a current set of *leaf* clusters corresponding to the current free energy minimum. To refine such a solution, we search for the lowest β that causes some leaf cluster to split. Ideally, there is just one split at that critical value, but for practical performance and numerical accuracy reasons we may have several splits at the new critical point. The splitting procedure can then be repeated to achieve the desired number of clusters or model cross-entropy.

```
β ← β0
create initial centroid

REPEAT until β = β_MAX or enough clusters:
    For each centroid c, create twin c*

    REPEAT until twins (c, c*) split or too many iterations:
        Estimate membership probs by (3.9)
        Estimate centroids by (3.14)

    IF more than one centroid split
    THEN [raised β too quickly]
        lower β
    ELSE IF no centroid split
        raise β
    ELSE [one centroid split]
        raise β
    delete extra twins c*
```

Figure 3.2: Clustering algorithm

3.4 Clustering Examples

> The properties that the child can detect in the input – such as the serial positions and adjacency and co-occurrence relations among words – are in general linguistically irrelevant. (Pinker, 1984, pg. 50)

In this section, we describe experiments with clustering words using the procedure described in the previous section. As explained there, our clustering procedure yields for each

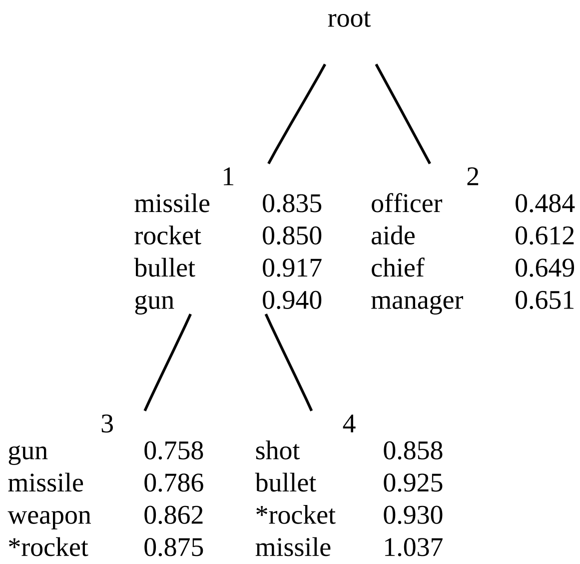

Figure 3.3: Direct object clusters for verb *fire*

value of β a set $\mathcal{C}_\beta$ of clusters minimizing the free energy F, with the model estimate for the conditional probability of a verb y given a noun x being

$$\tilde{P}(y|x) = \sum_{c \in \mathcal{C}_\beta} \tilde{P}(c|x)\tilde{P}(y|c),$$

where $\tilde{P}(c|x)$ depends on β. Recall that the pair (x, y) means that x occurred as the head noun of the direct object of verb y; for example, the pair (thesis,write) might be extracted from the sentence “You should write your thesis”.

In our first experiment, we wanted to choose a small set of nouns that we could be sure bore some relation to one another. Therefore, we chose the set $\mathcal{X}$ to consist of the 64 nouns appearing most frequently as heads of direct objects of the verb “fire” in the Associated Press newswire for 1988. In this corpus, the chosen nouns appeared as direct object heads of a total of 2147 distinct verbs, so each noun was represented by a density over 2147 verbs.

Figure 3.3 shows the five words most similar to the cluster centroid for the four clusters resulting from the first two cluster splits, along with the KL divergences from the centroids. It can be seen that the first split separates the objects corresponding to the weaponry sense of “fire” (cluster 1) from the ones corresponding to the personnel action (cluster 2). The second split then further refines the weaponry sense into a projectile sense (cluster 4) and a projector (of projectiles) sense (cluster 3). That split is somewhat less sharp, perhaps because not enough distinguishing contexts occur in the corpus. Notice that “rocket” is close to both centroids 3 and 4 and therefore has a high probability of belonging to *both* classes: our “soft” clustering scheme allows this type of ambiguity. Note that the “senses” we refer to are our own designations for the clusters – the algorithm does not decide what the sense(s) of a cluster actually are.

Our second experiment was performed on a bigger data set: we used object-verb pairs involving the 1000 most frequent nouns in the June 1991 electronic version of Grolier’s

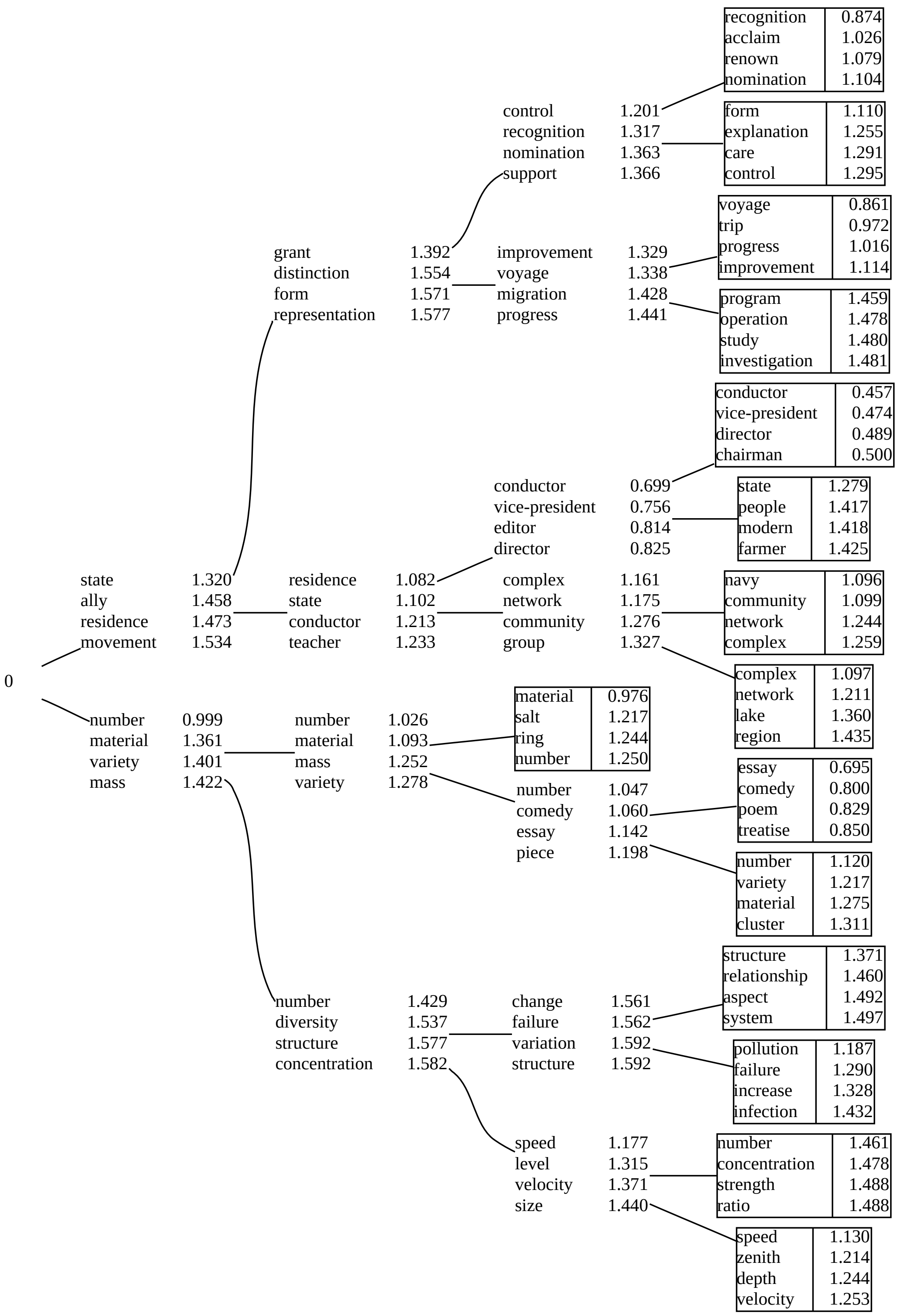

Figure 3.4: Noun clusters for Grolier's encyclopedia

Encyclopedia (10 million words). Figure 3.4 shows the four closest nouns for each centroid in a set of hierarchical clusters derived from this corpus. Again, we notice that the clusters and cluster splits often seem to correspond to natural sense distinctions. We also observe that a general word like "number" is close to quite a few cluster centroids.

3.5 Model Evaluation

The preceding qualitative discussion provides some indication of what aspects of distributional relationships may be discovered by clustering. However, we also need to evaluate clustering more rigorously as a basis for models of distributional relationships. We now look at two kinds of measurements of model quality: (i) KL divergence between held-out data and the asymmetric model, and (ii) performance on the task of deciding which of two verbs is more likely to take a given noun as direct object when the data relating one of the verbs to the noun has been withheld from the training data.

The evaluation described below was performed on a data set extracted from 44 million words of 1988 Associated Press newswire by using the pattern-matching techniques mentioned earlier. This collection process yielded 1112041 verb-object pairs. We then selected the subset involving the 1000 most frequent nouns in the corpus for clustering, and randomly divided it into a training set of 756721 pairs and a test set of 81240 pairs. Figure 3.5 shows the closest nouns to the cluster centroids in an early stage of the hierarchical clustering of the training data.

3.5.1 KL Divergence

Figure 3.6 plots the aggregate KL divergence of several data sets to cluster models of different sizes; the higher the KL divergence, the worse the coding inefficiency of using the cluster model. The aggregate KL divergence is given by

$$\sum_x D(P_{MLE}(\cdot|x)||\tilde{P}(\cdot|x)).$$

For each critical value of β, we show the aggregate KL divergence with respect to the cluster model based on C_β for three sets: the training set (set *train*), a randomly selected held-out test set (set *test*), and a set of held-out data for a further 1000 nouns that were not clustered (set *new*).

Not surprisingly, the training set aggregate divergence decreases monotonically. The test set aggregate divergence decreases to a minimum at 206 clusters and then starts increasing, which suggests that the larger models are overtrained.

The new noun test set is intended to evaluate whether clusters based on the 1000 most frequent nouns are useful classifiers for the selectional properties of nouns in general. We characterize each new noun x by its maximum likelihood distribution $P^{\mathrm{new}}_{MLE}(\cdot|x)$ as estimated from the new sample (we can't use the training data since the new nouns by definition don't appear there). The corresponding cluster membership probabilities for a new noun then have the form

$$\tilde{P}(c|x) = \exp\left(-\beta D(P^{\mathrm{new}}_{MLE}(\cdot|x)||P(\cdot|c))\right)/Z_x$$

and the model probability estimate is calculated as before. As the figure shows, the cluster

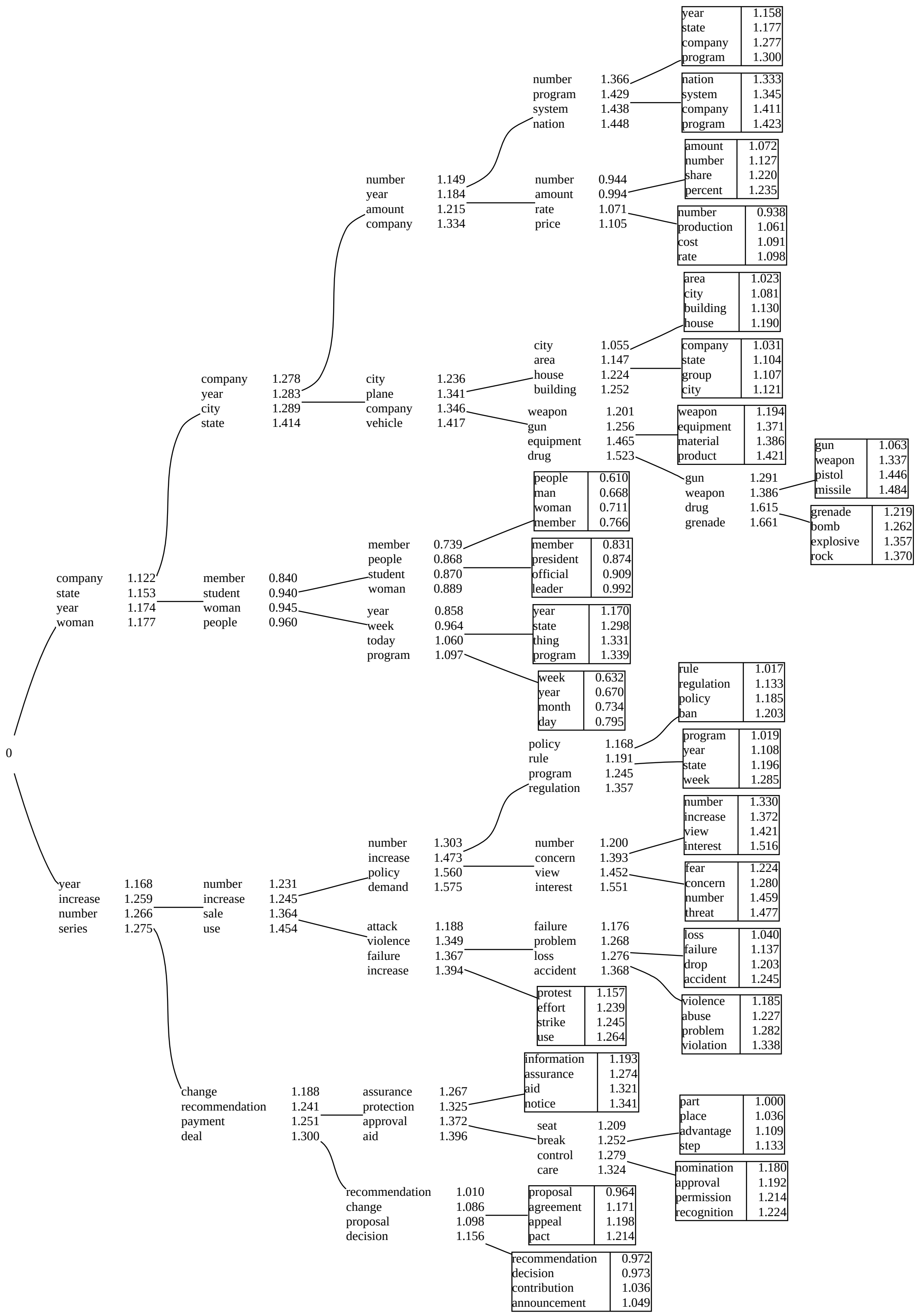

Figure 3.5: Noun clusters for 1988 Associated Press newswire

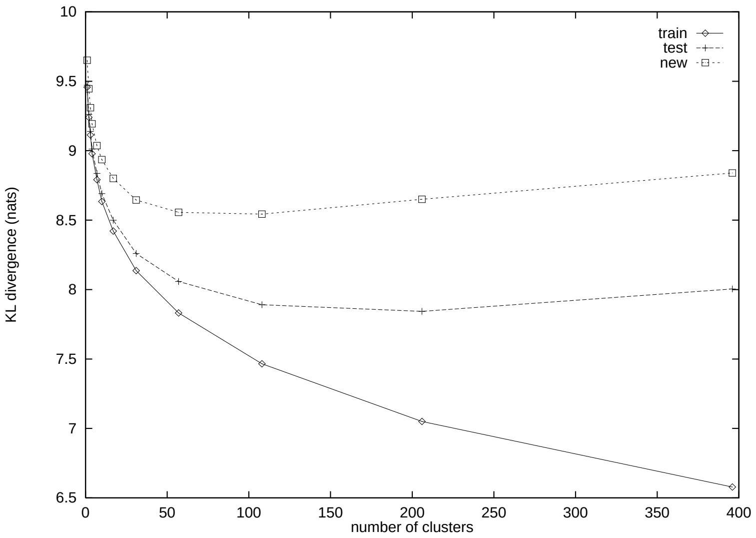

Figure 3.6: Model evaluation, 1988 Associated Press object-verb pairs

model provides over one nat of information about the selectional properties of the new nouns, although the overtraining effect is even more pronounced than for the held-out data involving the 1000 clustered nouns.

3.5.2 Decision Task

We also evaluated our cluster models on a verb decision task related to applications in disambiguation in language analysis. The task consists of judging which of two verbs y and y' is more likely to take a given noun x as object when all occurrences of (x, y) in the training set were deliberately deleted. Thus, this test evaluates how well the models reconstruct missing data from the cluster centroids, since we are interested in cluster models that can help solve sparse data problems.

The data for this test was built from the training data for the previous one in the following way, based on an experiment by Dagan, Marcus, and Markovitch (1995). We randomly picked 104 object-verb pairs (x, y) such that verb y appeared fairly frequently (between 500 and 5000 occurrences), and deleted all occurrences of such pairs from the training set. The resulting training set was used to build a sequence of cluster models as before. To create the test set, for each verb y in a deleted pair, a *confusion set* $\{y, y'\}$ was created. Then, each model was presented with the triple (y, x, y'), and was asked to decide which of y and y' is more likely to appear with a noun x.

Of course, we need some way of judging correctness without having access to the true pair probabilities, since the source distribution for natural language is presumably unknown. We fall back on the empirical frequencies to give us a rough estimate of the correct answer. Since these frequencies are known not to be entirely accurate (otherwise, we would have no need of cluster models!), we choose to create confusion sets for a noun x out of pairs of verbs y and y' such that one of the verbs occurred at least twice as often with x than the other in the original data set (prior to the pair deletion). Thus, we can be reasonably sure that whichever verb occurred with x more often in the training set truly has a higher probability of co-occurrence.

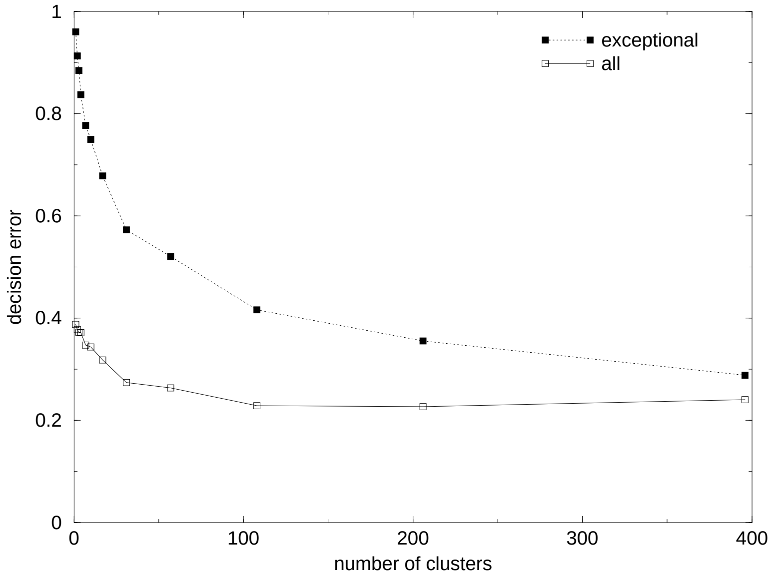

Figure 3.7: Pairwise verb comparisons, 1988 Associated Press object-verb pairs

In order to evaluate performance, we compare the sign of $\log\left(\tilde{P}(y|x)/\tilde{P}(y'|x)\right)$ with that of $\log\left(P_{MLE}(y|x)/P_{MLE}(y'|x)\right)$ on the initial data set. The error rate for each model is simply the proportion of sign disagreements over the test corpus. Figure 3.7 shows the error rates for each model on all the selected (y, x, y') (*all*) and for just those *exceptional* triples in which the log frequency ratio of (x, y) and (x, y') differs from the log marginal frequency ratio of y and y'.

The exceptional cases are especially interesting in that estimation methods (such as Katz's back-off method) based just on the marginal frequencies, which the initial one-cluster model represents, would be consistently wrong. We see that the cluster model tremendously outperforms classic estimation methods in the exceptional cases, and thus has the potential to provide a much better solution to the sparse data problem. Furthermore, while some overtraining effects can be observed for the largest models considered, these effects do not appear for the exceptional cases.

3.6 Related Work

It is beyond the scope of this thesis to provide a review of the entire body of clustering literature; data clustering has been discussed in fields ranging from statistics to biology. One list of journals that publish papers on the subject contains 987 entries (Classification Society of North America, 1996); indeed, a summary of various clustering methods is a thesis in itself (Anderberg, 1973, "substantially this same text was submitted as a dissertation", pg. xiii). We therefore narrow our focus to two subjects: clustering methods appearing in the natural language-processing literature (section 3.6.1), and other probabilistic clustering methods (section 3.6.2).

3.6.1 Clustering in Natural Language Processing

Quite a few methods for distributional clustering have appeared in the literature of the natural language processing community, although to the best of our knowledge, our work is the first to use soft clustering in a language-processing context. The algorithms we will describe here algorithms fall into two categories: those that seek to find classes corresponding to human concepts, and those that create classes for the purpose of improving language modeling.

As an aside, we note that these two categories correspond to two orthogonal trends in clustering work in general. The first trend, readily apparent in recent work on data mining and knowledge discovery, is to find clusters that are somehow well-formed. Work in this vein uses optimization criteria concerning cluster structure; for instance, our distortion function $\mathcal{D}$ measures the average distance between objects and centroids. The other trend is to find clusters that aid in the performance of some task; work in this area uses optimization criteria based on likelihood or some other performance measure.

Clustering for Clusters' Sake

Most of the methods whose end goal is the production of clusters (and therefore do not test whether the clusterings aid in the performance of some task) are geared towards finding either semantic or syntactic classes. The work of Hatzivassiloglou and McKeown (1993) (henceforth H&M) is notable because they provide a way to evaluate the goodness of semantic clusterings; many other papers (for example, Finch and Chater (1992) or Schütze (1993)) merely present example clusters and state that the derived classes seem to correspond to intuition.

H&M describe a hard clustering scheme for grouping semantically-related adjectives. They treat adjectives as distributions over the nouns they modify, and use Kendall's τ coefficient (studied in section 2.3.4) to measure the distance between these distributions. Their optimization function is one of well-formedness: it rewards partitions that minimize the average distance between adjectives in the same cluster. They carefully delineate a rigorous evaluation method for comparing clusterings produced by their algorithm against clusterings produced by human judges,[2] computing precision, recall, fallout and F-measure results with respect to an average of the responses given by the judges, thereby taking into account the fact that humans do not always agree with each other.

An interesting feature of their work is that they incorporate negative linguistic similarity information. By simply observing that adjectives in the same noun phrase should not, for a variety of linguistic reasons, be placed in the same class, they get dramatically better results (17-50% improvement across the various performance metrics).

Some superficial similarities with our clustering work are readily apparent. The distributional similarity component of H&M's system treats adjectives as distributions over nouns, while we treat nouns as distributions over verbs. Also, we and H&M both used Associated Press newswire as training data, although H&M only used 8.2 million words, as opposed to our 44 million. However, our results are incomparable because our goals differ. H&M explicitly aim to create classes of semantically related words, and so must solicit human judgments. They were therefore constrained by human limitations to clustering

[2] One minor criticism: the number of clusters to create was a parameter given to the system, whereas the humans were free to choose whatever number of clusters they wished.

only 21 adjectives. We, on the other hand, are more interested in clusterings that improve performance and so make use of a great deal more data.

An independent body of work seeking to build classes corresponding to human intuitions is the field of language clustering. Many researchers in comparative lexicostatistics study the problem of how to create hierarchical clusterings that correspond to the evolution and splitting off of languages over time. Black and Kruskal (1997) give a short history and bibliography of the field.

Clustering for Language Modeling

A large number of papers have been written on using class-based models to improve language modeling (five such papers appear in the 1996 ICASSP proceedings alone (Sig, 1996)). A common approach is to group words by their parts of speech. However, there is no reason to believe that classifications based on parts of speech are optimal with respect to language modeling performance, so we look at papers which present novel clustering techniques. Since the methods we discuss all attempt to create probabilistic models with strong predictive power, it is not surprising that they are all guided by the maximum likelihood principle.

The most well-known class-based method is the work by Brown et al. (1992). In their setting, the set of objects and the set of contexts are the same ($\mathcal{X} = \mathcal{Y} = \mathcal{W}$); the pair (w_1, w_2) denotes the appearance of the two-word sequence $w_1 w_2$ in the training sample. Brown et al. assume a Boolean clustering of the data, so that each word w belongs only to the class $c(w)$, where $c(\cdot)$ is the membership function. Then, their class-based probability estimate takes the form

$$\tilde{P}(w_2|w_1) = \tilde{P}(w_2|c(w_2))\tilde{P}(c(w_2)|c(w_1)). \tag{3.16}$$

Given the membership function, the parameters $\tilde{P}(w|c(w))$ and $\tilde{P}(c(w_2)|c(w_1))$ are determined by sample frequencies, so only the function $c(\cdot)$ needs to be estimated. This is done by attempting to find class assignments that maximize the average mutual information $\langle I \rangle$ of the clusters, which in the limit is equivalent to maximizing the likelihood: if $t_1 t_2 \ldots t_n$ is the training text, then

$$\begin{aligned} L_c &= \frac{1}{n-1} \log \tilde{P}(t_2 \ldots t_n | t_1) \\ &\approx -H(t) + \langle I \rangle, \end{aligned}$$

where $H(t)$ is the entropy of the unigram (single word) distribution, which we can consider to be fixed.

A serious problem Brown et al. face is that they do not have a way to calculate good estimates for $c(\cdot)$. Therefore, in each iteration step of their agglomerative clustering algorithm, they are forced to try many different merges of classes to find the one yielding the best improvement in $\langle I \rangle$. After some amount of care, they are able to derive an algorithm that takes $O(|\mathcal{W}|^3)$ time in each iteration step, whereas in the same setting our iteration steps would take $O(|\mathcal{C}||\mathcal{W}|^2)$ time, which is a significant savings if the number of clusters is small relative to the number of words. Also, once the desired number of clusters has been achieved, Brown et al. shift words from cluster to cluster in order to compensate for premature groupings of words in the same class – this is the rigidity problem referred to in the quotation from Kaufman and Rousseeuw earlier in this chapter. Since we create a soft clustering, we never have to compensate for words being incorrectly classed together. At any rate, Brown et al.'s method potentially involves much wasted computation since both good and bad merges and shifts must be tried, whereas we are guaranteed that each step we take reduces the free energy.

Brown et al. do present an alternative algorithm which spends $O(|\mathcal{C}|^3)$ time in each iteration. This algorithm sorts the words by frequency and puts the top k into their own classes. Each iteration step consists of adding the next most frequent word yet to be clustered as a new class and then finding the best merge among the new set of classes; when this merge is taken, the system once again has k clusters. On the other hand, it is possible that this heuristic narrows the search down so much that good classings are missed. This may well explain the small perplexity reduction achieved by Brown et al.'s method on the Brown corpus (from 244 to 236 using a model that interpolates the class-based model with word-based estimators).

Another commonly-cited class-based language-modeling method, that of Kneser and Ney (1993), is presented by Ueberla (1994). In many respects, Kneser and Ney's work is quite similar to that of Brown et al. The same probability model (3.16) is used, and some of the heuristics employed to speed up calculations are the same as well. However, their optimization criterion differs, although it, too, is derived via the maximum likelihood principle. Instead of an agglomerative clustering algorithm, Kneser and Ney start with the desired number of clusters, so that the only operation undertaken to improve the clustering is to move words from one cluster to another, searching for the move that makes the biggest improvement. The running time of each such iteration step is $O(|\mathcal{W}| \cdot (|\mathcal{W}| + |\mathcal{C}|^2))$.

Ueberla reports that Kneser and Ney's method achieves perplexity improvements of up to 25% on Wall Street Journal data with respect to Katz's back-off method. This is a rather stunning result. However, the class-based model uses a smoothing method known as absolute discounting (Ney and Essen, 1993). An interesting question is how much of the performance is due to the smoothing method and how much is due to the clustering (Brown et al. did not smooth the data); no comparison was done between the class-based method and the absolute discounting method.

3.6.2 Probabilistic Clustering Methods

One of the first papers to discuss the notion of probabilistic clustering is that of Ruspini (1970), who was inspired by Zadeh's work on fuzzy sets (Zadeh, 1965). His method attempts to find membership probabilities (which he calls "degrees of belongingness") that optimize certain well-formedness conditions similar to our distortion function $\mathcal{D}$. However, he does not attempt to mathematically derive good estimates, so the search for good parameter settings consists of repeatedly altering one membership probability while keeping all the others fixed. Furthermore, his method relies on distances between objects, rather than between objects and average distributions. This poses no problem in his case because he only considers artificial problems where the true distances are known. In practice, however, estimates of inter-object distances can be quite sensitive to noise; centroid methods overcome this problem by averaging together many points.

The *fuzzy k-means* method (Bezdek, 1981), a generalization of the k-means approach, bears some resemblance to our procedure. It is a centroid method using the Euclidean distance (L_2) as distance function. The centroid distributions depend on the squares of the membership probabilities:

$$\tilde{P}(y|c) = \frac{\sum_x \left(\tilde{P}(c|x)\right)^2 P_{MLE}(y|x)}{\sum_x \left(\tilde{P}(c|x)\right)^2},$$

and the membership probabilities in turn depend on the positions of the centroids. The optimization function rewards clusterings that minimize the distance between objects and centroids:

$$F_{OPT} = \sum_x \sum_c \left(\tilde{P}(c|x)\right)^2 (L_2(x,c))^2.$$

This is a well-formedness condition rather than a maximum likelihood criterion, and in fact fuzzy k-means is not meant to produce probability estimates. It is also not meant to produce a hierarchical clustering; the number of centroids is kept constant throughout the iterated estimation process.

The clustering procedures most similar to our own are the deterministic annealing approaches; these include the work of Rose, Gurewitz, and Fox (1990) (which influenced our approach) and Hofmann and Buhmann (1997). These both find clusters that minimize the free energy (3.5). An important difference is that they use the squared Euclidean distance (L_2), whereas we use the KL divergence as distance function. In the distributional setting we have been considering, using the KL divergence is well-motivated, whereas it is not entirely clear why the L_2 norm would be meaningful.

Bayesian methods (Wallace and Dowe, 1994; Cheeseman and Stutz, 1996) combine well-formedness constraints and performance criteria. They seek to find the model with the maximum posterior probability given the data, where the posterior probability is based on the product of the model prior and the likelihood the model assigns to the data. The prior is based on the structure of the cluster system, and in general encodes a bias for fewer clusters; such a prior serves to balance out the tendency of maximum likelihood criteria to reward systems that have a large number of clusters. This is analogous to our inclusion of a maximum entropy condition in the derivation of our method, since the maximum entropy criterion also tends to favor having fewer clusters.

The methods of Wallace and Dowe (1994) and Cheeseman and Stutz (1996) do not yield cluster hierarchies because the number of clusters is allowed to fluctuate from iteration step to iteration step. The "class hierarchy" described by Hanson, Stutz, and Cheeseman (1991) does not consist of classes but rather of attributes: each node in the dendrogram represents a collection of parameter settings inherited by all the descendents of that node.

3.7 Conclusions

We have described a novel clustering procedure for probability distributions that can be used to group words according to their participation in particular grammatical relations with other words. Our method builds a hierarchy of probabilistic classes using an iterative algorithm reminiscent of EM. The resulting clusters are intuitively informative, and can be used to construct class-based word coocurrence models with substantial predictive power.

While the clusters derived by the proposed method seem in many cases semantically significant, this intuition needs to be grounded in a more rigorous assessment. In addition to evalutions of predictive power of the kind we have already carried out, it might be worthwhile to compare automatically-derived clusters with human judgements in a suitable experimental setting, perhaps the one suggested by Hatzivassiloglou and McKeown (1993). In general, however, the development of methods for directly measuring cluster quality is an open research area; the problem is compounded when one takes hierarchical clusterings into account.

Another possible direction to take would be to move to other domains. For instance, document clustering has been studied by many researchers in the field of information retrieval. Recently, there has been renewed interest in using document clustering as a *browsing* aid rather than a search tool (see Cutting et al. (1992) for a short discussion), and also as a way to organize documents (Yahoo! (1997) provides a hierarchical clustering in which documents can appear in more than one class). In situations where $|\mathcal{X}|$ and $|\mathcal{Y}|$ are very large, our clustering algorithm may be somewhat slow; however, the extra descriptive power provided by our *probabilistic* clustering may well be worth the extra computational effort.

Chapter 4

Similarity-Based Estimation

In the previous chapter we looked to cluster centroids as a source of information when data on a specific event $\mathcal{E}$ was lacking. This chapter introduces an alternative model, where we instead look to the events most similar to $\mathcal{E}$. For convenience, we will refer to the new type of model as *similarity-based*, although our clustering method of the preceding chapter also made use of the notion of similarity.

4.1 Introduction

In the previous chapter, we described a method for automatically clustering distributional data, and showed that we can use the clusters so derived to construct effective models for predicting probabilities in situations where data is lacking. The clustering method was divisive: the system started with just one cluster centroid, and as the temperature was slowly lowered, phase transitions caused cluster centroids to split. This splitting of centroids meant that the number of clusters k did not have to be determined beforehand; rather, all possible numbers of clusters could be considered in a fairly efficient fashion, and the best configuration could be chosen via cross-validation. For clustering algorithms that keep the number of clusters constant throughout the estimation process, the only way to try out many different numbers of clusters is to re-run the algorithm with a different value of k each time. But it is generally the case for these algorithms that the results of the computation for one k cannot be used to aid the computation for a different k, so the search for the right k is not very efficient.

An interesting alternative is a nearest-neighbor approach, where given an event $\mathcal{E}$ whose probability we need to estimate, we consult only those events that are most similar to $\mathcal{E}$. In a sense, we allow each event to form the centroid of its own class, and thus avoid having to find the right number of clusters. While this approach does not reduce the size of the model parameter space as class-based approaches do, it avoids the over-generalization that class-based models can fall prey to. As Dagan, Marcus, and Markovitch (1995) argue, using class information to model specific events may lead to too much loss of information. Probabilistic clusterings ameliorate this problem somewhat by combining estimates from different classes, using membership probabilities to weight the class estimates appropriately, but the concern about over-generalization is still valid.

We present a small example to make the over-generalization problem clearer. Figure 4.1 shows a situation in which there are four objects (the empty circles) and only one centroid (the grey circle in the middle). Let us assume we are trying to model the behavior of X. In

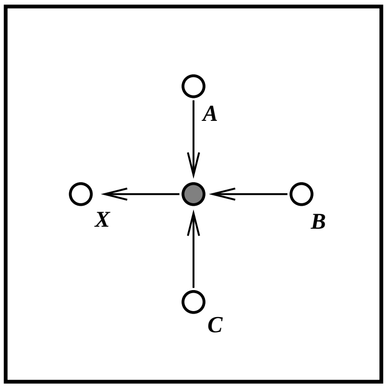

Figure 4.1: Centroid overgeneralization

a cluster-based centroid model, the estimate for the behavior of X depends on the behavior of the centroid; this dependence is indicated by the arrow from the centroid to X. However, the behavior of the centroid is an average of the behaviors of all the other points, including A, B, and C, as indicated by the arrows pointing to the centroid. Therefore, an estimate for X depends not only on A and C, which are relatively close to it, but also on B, which is much farther way. It might make more sense to use only points A and C in trying to estimate X.

We therefore turn our attention in this and the next chapter to similarity-based language modeling techniques that do not require building general classes. While our cluster model, described in the previous chapter, estimated the conditional probability $P(y|x)$ of an object-context pair by averaging together class estimates $P(y|c)$, weighting the evidence of each class by the degree of association $P(c|x)$ between x and c:

$$\tilde{P}(y|x) = \sum_{c \in \mathcal{C}} \tilde{P}(c|x)\tilde{P}(y|c),$$

our new object-centered model replaces the centroids by other objects:

$$\tilde{P}(y|x) = \sum_{x'} f(x, x')P(y|x'), \tag{4.1}$$

where $f(x, x')$ depends on the similarity between x and x'.

We are not the originators of equation (4.1). Similarity-based estimation was first used for language modeling in the *cooccurrence smoothing* method of Essen and Steinbiss (1992), derived from work on acoustic model smoothing by Sugawara et al. (1985). Karov and Edelman (1996) develop a similarity-based disambiguation method that also can be fit into the framework of equation (4.1); however, since their method does not estimate probabilities and relies on a similarity function that is calculated via an iterative process, we will not give further consideration to their work here.

In this chapter we establish proof of concept: we discuss and compare ways to instantiate equation (4.1), using a simple decision task for evaluation purposes. The KL divergence will once again prove to be an effective measure of dissimilarity. In the next chapter we evaluate a similarity-based model on more true-to-life tasks that test the utility of our method for speech recognition; we use a more complicated version of the model presented here, incorporating several heuristics in order to speed up the computation.

4.2 Chapter Overview

As in the previous chapter, our goal is to estimate the (conditional) probability of object-context pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Our first concern in this chapter is to describe similarity-based estimation methods in general. In section 4.3 we develop a common framework for these methods, so that the only parameter that varies from method to method is the similarity function used. In the following section (4.4) we describe various similarity functions. The majority are based on distance functions studied in chapter 2, but we also discuss the *confusion probability*, which appears in the work of Essen and Steinbiss (1992).

The second part of this chapter describes our evaluation of similarity-based methods. In section 4.5.1, we introduce the problem of *pseudo-word disambiguation*, a task which is related to the usual word sense disambiguation problem, but presents many advantages in terms of ease of experimentation. After a discussion of the data used to construct basic language models and a comparison of these basic models (4.5.2), we look at a few examples to get a qualitative sense for how the different similarity functions perform (4.5.3). Finally, section 4.5.4 presents five-fold cross-validation results for the similarity-based methods and for several baseline models. Our tests show that indeed, similarity information can be quite useful in sparse data situations. In particular, we found that all the similarity-based methods performed almost 40% better than back-off if unigram frequency was eliminated from being a factor in the decision.

An interesting phenomenon we observe is that the effect of removing extremely rare events from the training set is quite dramatic when similarity-based methods are used. We found that, contrary to a claim made by Katz that such events can be discarded without hurting language model performance (Katz, 1987), similarity-based smoothing methods suffer noticeable performance degradation when singletons (events that occur exactly once) are omitted.

Throughout this chapter, the base of the logarithm function is 10.

4.3 Distributional Similarity Models

A similarity-based language model consists of three parts: a scheme for deciding when to use similarity-based information to determine the probability of a word pair, a method for combining information from similar words, and, of course, a function measuring the similarity between words. We give the details of each of these three parts in the following three sections.

4.3.1 Discounting and Redistribution

We hold that it is best to always use the most specific information available. While the maximum likelihood estimate (MLE)

$$P_{MLE}(y|x) = \frac{C(x, y)}{C(x)}$$

(equation (2.1) from chapter 2, where $C(z)$ is the number of times event z occurred in the training data) yields a terrible estimate in the case of an unseen word pair, it is pretty good when sufficient data exists. Therefore, Katz's (1987) implementation of the Good-Turing discounting method, described in chapter 2, provides an attractive framework for similarity-based methods; it uses the (discounted) MLE when the pair (x, y) occurs in the data, and a different estimate if the pair does not occur:

$$\hat{P}(y|x) = \begin{cases} P_d(y|x) & \text{if } C(x,y) > 0 \\ \alpha(x)P_r(y|x) & \text{otherwise } ((x,y) \text{ is unseen}) \end{cases} . \tag{2.3}$$

Recall that equation (2.3) actually represents a modification of Katz's formulation: we have written $P_r(y|x)$ where Katz has $P(y)$. This allows us to use similarity-based estimates for unseen word pairs, rather than simply backing off to the probability of the context y. Observe that this formulation means that we will use the similarity estimate for unseen word pairs only, as desired.

We next investigate estimates for $P_r(y|x)$ derived by averaging information from objects that are distributionally similar to x.

4.3.2 Combining Evidence

The basic assumption of a similarity-based model is that if object x' is "similar" to object x, then the behavior of x' can yield information about the behavior of x. When data on x is lacking, then, we average together the distributions of similar objects, weighting the information furnished by a particular x' by the similarity between x' and x.

More precisely, let $W(x, x')$ denote an increasing function of the similarity between x and x'; that is, the more similar x and x' are, the larger $W(x, x')$ is. Let $\mathcal{S}(x)$ denote some set of objects that are most similar to x (we discuss the exact form of $\mathcal{S}(x)$ in the next paragraph). Then, the general form of similarity model we consider is a W-weighted linear combination of predictions of similar objects:

$$P_{SIM}(y|x) = \sum_{x' \in \mathcal{S}(x)} \frac{W(x, x')}{\sum_{x' \in \mathcal{S}(x)} W(x, x')} P(y|x'). \tag{4.2}$$

Observe that according to this formula, we predict that y is likely to occur with x if it tends to occur with objects that are very similar to x.

Considerable latitude is allowed in defining the set $\mathcal{S}(x)$. Essen and Steinbiss (1992) and Karov and Edelman (1996) (implicitly) set $\mathcal{S}(x) = \mathcal{X}$. However, if $\mathcal{X}$ is very large, it is desirable to restrict $\mathcal{S}(x)$ in some fashion, so that summing over all $x' \in \mathcal{X}$ is not too time-consuming. In the next chapter, we will consider various heuristics for choosing a small set of similar words. These heuristics include setting a limit on the maximum size of $\mathcal{S}(x)$, and only allowing an object x' to belong to $\mathcal{S}(x)$ if the dissimilarity between x and x' is less than some threshold value. We will show some evidence at the end of this chapter that limiting the size of the set of closest objects does not greatly degrade performance, at least for the best similarity-based models.

The approach taken in this chapter is to use P_{SIM} as the probability redistribution model in equation (2.3), i.e., $P_r(y|x) = P_{SIM}(y|x)$. In the next chapter we discuss a variation in which P_r is a linear combination of P_{SIM} and another estimator.

4.4 Similarity Functions

The final step in defining a similarity-based model is to choose what similarity function to use. We first look in section 4.4.1 at three functions from chapter 2 that measure the distance between distributions. For each of these functions, it is necessary to define a weight function $W(x, x')$ that "reverses the direction" of the distance function, since we need weights that have larger values when the distributions are less distant. Section 4.4.2 describes some of the properties of the *confusion probability*, which was used to achieve good performance results by Essen and Steinbiss (1992). Section 4.4.3 discusses the base language models from which object distributions are computed, and also summarizes some properties of the four similarity functions we will compare.

Regardless of which similarity function is chosen, in order to make the computation of equation (4.2) efficient it is useful to compute the $|\mathcal{X}| \times |\mathcal{X}|$ matrix of similarities $W(x_i, x_j)$ or distances $d(x_i, x_j)$ (for arbitrary distance functions d) beforehand.

4.4.1 Distance Functions

In chapter 2, we studied several functions measuring the distance between probability distributions. These included the KL divergence (section 2.3.1)

$$D(x||x') = \sum_y P(y|x) \log \frac{P(y|x)}{P(y|x')},$$

the total divergence to the mean (section 2.3.2)

$$A(x, x') = D(x||\frac{x + x'}{2}) + D(x'||\frac{x + x'}{2})$$

($(x + x')/2$ denotes the probability mass function $(P(\cdot|x) + P(\cdot|x'))/2$), and the L_1 norm (section 2.3.3)

$$L_1(x, x') = \sum_y |P(y|x) - P(y|x')|.$$

Since these functions are all distance functions, they *decrease* when the similarity between x and x' increase. However, we desire weight functions $W(x, x')$ that are *increasing* in the similarity between x and x'.

In the case of the KL divergence D, we set $W(x, x')$ to be

$$W_D(x, x') = 10^{-\beta D(x||x')}.$$

β is an experimentally-tuned parameter controlling the relative influence of the objects closest to x: if β is high, then $W(x, x')$ is non-negligible only for those x' that are extremely close to x, whereas if β is low, objects that are somewhat distant from x also contribute to the estimate. The choice of a negative exponential form is motivated by the fact that the probability of drawing an i.i.d. sample of size n with empirical distribution P from a multinomial Q is $10^{-nD(P||Q)}$ to first order in the exponent – this is theorem 2.2 from section 2.3.1.

When the distance function is the total divergence to the mean, we also use a negative exponential:

$$W_A(x, x') = 10^{-\beta A(x, x')}.$$

Again, β controls the relative importance of the most similar objects and is determined experimentally.

Finally, we define the weight function for the L_1 norm to be

$$W_{L_1}(x, x') = (2 - L_1(x, x'))^{\beta},$$

with β playing the same role as in W_D and W_A above (we tried using the exponential form $10^{-\beta L_1(x,x')}$, but $(2 - L_1(x, x'))^{\beta}$ yielded better performance results).

We have made no attempt to normalize these various weight functions, so they take on different sets of values; for example, $W_D(x, x) = W_A(x, x) = 1$, but $W_L(x, x) = 2^{\beta}$. Normalization is not necessary because our evaluation task ignores scale factors.

4.4.2 Confusion probability

Essen and Steinbiss (1992) introduced *confusion probability* in the context of cooccurrence smoothing for language modeling. Cooccurrence smoothing was also applied by Grishman and Sterling (1993) to the problem of estimating the likelihood of selectional patterns.

Of the four similarity-based models Essen and Steinbiss consider, we choose to describe and implement model 2-B (equivalent to model 1-A) because it was found to be the best performer of the four. Indeed, Essen and Steinbiss report test-set perplexity reductions of up to 14% on small corpora. Although they used an interpolation framework, where the similarity-based estimate was linearly interpolated with other estimators for seen as well as unseen events, we will for the sake of uniformity incorporate the confusion probability into the back-off-like framework of equation (2.3).

The confusion probability represents the likelihood that object x' can be substituted for object x; it is based on the probability that x and x' are found in the same contexts:

$$P_C(x'|x) = \sum_y \frac{P(x|y)P(x'|y)P(y)}{P(x)} \tag{4.3}$$

(the term $P(x)$ is required to ensure that $\sum_{x'} P_C(x'|x) = 1$). Since this expression incorporates both conditional probabilities and marginal probabilities, it is not a measure of the distance between two distributions as are the functions described in section 2.3.

The confusion probability is symmetric in the sense that $P_C(x'|x)$ and $P_C(x|x')$ are identical up to frequency normalization: $\frac{P_C(x'|x)}{P_C(x|x')} = \frac{P(x)}{P(x')}$. Unlike the measures described above, x may not be the "closest" object to itself, that is, there may exist an object x' such that $P_C(x'|x) > P_C(x|x)$, as we shall see in section 4.5.3.

Further insight into the behavior of P_C is gained by using Bayes' rule to rewrite expression (4.3):

$$\begin{aligned} P_C(x'|x) &= \sum_y \frac{1}{P(x)} \left(\frac{P(y|x)P(x)}{P(y)}\right) \left(\frac{P(y|x')P(x')}{P(y)}\right) P(y) \\ &= \sum_y \frac{P(y|x)}{P(y)} P(y|x')P(x'). \end{aligned}$$

This form reveals another important difference between the confusion probability and the functions D, A, and L_1 described above. The latter three functions rate x' as similar to x if, roughly, $P(y|x')$ is high when $P(y|x)$ is. $P_C(x'|x)$, however, is greater for those x' for which

$P(x', y)$ is large when $P(y|x)/P(y)$ is. Notice that the case when the ratio $P(y|x)/P(y)$ is large contradicts the back-off assumption that $P(y)$ is a good estimate of $P(y|x)$ when the pair (x, y) is unseen.

While the fact that P_C is called a probability implies that it ranges between 0 and 1, some elementary calculations show that in fact its maximum value is $\frac{1}{2} \max_y P(y)$. Following Essen and Steinbiss, we choose the weight function $W(x, x')$ to be the confusion probability itself without including the scale parameter β.

4.4.3 Base Language Models

Throughout the above discussion, we have blithely referred to the quantities $P(y|x)$, $P(x)$, and $P(y)$ without explaining where these quantities actually come from. These must be provided by some base language model P, but it turns out that there is some subtlety as to the form the base language model may take.

As discussed in section 2.3.1, the KL divergence $D(x||x')$ is undefined if there exists a context y such that $P(y|x)$ is greater than zero but $P(y|x')$ is zero. This argues for a language model that is smoothed so that $P(y|x')$ cannot be zero. A natural choice is to use the back-off estimate, so that $P(y|x) = P_{BO}(y|x)$, where P_{BO} is given by equation (2.4).

However, the normalization of the confusion probability (4.3) requires that the base language model be consistent with respect to joint and marginal probabilities, that is, that

$$P(x) = \sum_y P(y|x)P(x).$$

Unfortunately, the back-off estimate does not have this property, since it discounts conditional probabilities without altering the marginals. Therefore, we use the maximum likelihood estimate as the base language model for P_C: $P(y|x) = P_{MLE}(y|x)$

Thus, we cannot directly compare the performances of all four of the similarity-based models defined above because they require different base language models. In the experimental results section of this chapter, then, we will evaluate the total divergence to the mean, the L_1 norm, and the confusion probability, using P_{MLE} as the base language model. Chapter 5 describes experiments where the KL divergence is used as the distance function and the back-off estimate is used as the base language model.

Several features of the measures of similarity listed above are summarized in table 4.1. "Base LM constraints" are conditions that must be satisfied by the probability estimates of the base language model. The last column indicates whether the weight $W(x, x')$ associated with each similarity function depends on a parameter that needs to be tuned experimentally.

distance	range	base LM constraints	tune?
D	$[0, \infty]$	$P(y|x') \neq 0$ if $P(y|x) \neq 0$	yes
A	$[0, 2\log 2]$	none	yes
L_1	$[0, 2]$	none	yes
P_C	$[0, \frac{1}{2}\max_y P(y)]$	Bayes consistency	no

Table 4.1: Summary of similarity function properties

4.5 Experimental Results

We evaluated three of the similarity measures described above on a word sense disambiguation task. Each method is presented with a noun and two verbs, and must decide which verb is more likely to have the noun as a direct object. Thus, we do not measure the absolute quality of the assignment of probabilities, as would be the case in a standard language model evaluation such as perplexity reduction (defined in the next chapter) but merely ask that a method be able to distinguish between two alternatives. We are therefore able to ignore constant factors, and so need neither normalize the similarity measures to lie between 0 and 1 nor calculate the denominator in equation (4.2).

4.5.1 Pseudo-word Sense Disambiguation

In the usual word sense disambiguation task, the method to be tested is presented with an ambiguous word in some context, and is asked to use the context to identify the correct sense of the word. For example, a test instance might be the sentence fragment "robbed the bank"; the disambiguation method must decide whether "bank" refers to a river bank, a savings bank, or perhaps some other alternative.

While sense disambiguation is clearly an important task, it presents numerous experimental difficulties. First of all, the very notion of "sense" is not clearly defined; for instance, dictionaries may provided sense distinctions that are too fine or too coarse for the data at hand. Also, one needs to have training data for which the correct senses have been assigned, which can require considerable human effort.

To circumvent these and other difficulties, we set up a pseudo-word disambiguation experiment (Schütze, 1992; Gale, Church, and Yarowsky, 1992), the general format of which is as follows. We first construct a list of *pseudo-words*, each of which is the combination of two different words in $\mathcal{Y}$. Each word in $\mathcal{Y}$ contributes to exactly one pseudo-word. Then, we replace each y in the test set with its corresponding pseudo-word. For example, if we choose to create a pseudo-word out of the words "make" and "take", we would change the test data like this:

make plans $\Rightarrow$ {make, take} plans
take action$\Rightarrow$ {make, take} action

The method being tested must choose between the two words that make up the pseudo-word.

The pseudo-word set-up has two attractive features. First, the alternative "senses" are under the control of the experimenter. Each test instance presents exactly two alternatives to the disambiguation method, and the alternatives can be chosen to be of the same frequency, the same part of speech, and so on. Secondly, the pre-transformation data yields the correct answer, so that no hand-tagging of the word senses is necessary. These advantages make pseudo-word experiments an elegant and simple means to test the efficacy of different language models.

4.5.2 Data

We ran our evaluation on the same Associated Press newswire data that we used for the clustering evaluation described in the previous chapter. To review, we set $\mathcal{X}$ to be the 1000 most frequent nouns in the data; $\mathcal{Y}$ was the set of transitive verbs y that were observed

Training data for base language models		
with singletons:	587833	
without singletons:	505426	
Parameter tuning and test data		
T_1:	3434	(tuning: 13718)
T_2:	3434	(tuning: 13718)
T_3:	3434	(tuning: 13718)
T_4:	3434	(tuning: 13718)
T_5:	3416	(tuning: 13736)

Table 4.2: Number of bigrams in the training, parameter tuning, and test sets.

to take a noun in $\mathcal{X}$ as direct object. The extraction of object-verb pairs was performed via regular pattern matching and concordancing tools (Yarowsky, 1992a) from 44 million words of 1988 Associated Press newswire, which had been automatically tagged with parts of speech (Church, 1988). Admittedly, regular expressions are inadequate for this task; although we filtered the results somewhat, some bad pairs doubtless remained.

We used 80%, or 587833, of the pairs so derived, for building base bigram language models, reserving 20% for testing purposes. As some of the similarity measures to be compared require smoothed language models, while others do not, we calculated both a Katz back-off language model ($P = P_{BO}$) and a maximum likelihood model ($P = P_{MLE}$). Furthermore, we wished to investigate Katz's claim that one can delete *singletons*, word pairs that occur only once, from the training set without affecting model performance (Katz, 1987); our training set contained 82407 singletons. We therefore built four base language models, summarized in table 4.3.

	with singletons (587833 pairs)	omit singletons (505426 pairs)
MLE	MLE-1	MLE-o1
back-off	BO-1	BO-o1

Table 4.3: Base language models

Since we wished to test the effectiveness of using similarity information for unseen word cooccurrences, we removed from the test set any object-verb pairs that occurred in the training set; this resulted in 17152 *unseen* pairs (some occurred multiple times). The unseen pairs were further divided into five equal-sized parts, T_1 through T_5, which formed the basis for five-fold cross-validation: in each of the five runs, one of the T_i was used as a performance test set, with the other 4 sets combined into one set used for tuning parameters (if necessary) via a simple grid search. Finally, test pseudo-words were created from pairs of verbs with similar frequencies, so as to control for word frequency in the decision task. Our measure of performance was the *error rate*, defined as

$$\frac{1}{n}(\text{number of incorrect choices } + (\text{number of ties})/2)$$

where n was the size of the test corpus. A tie occurs when the two words making up a pseudo-word are deemed equally likely.

We first look at the performance of the base language models themselves. Their error rates are summarized in table 4.4. MLE-1 and MLE-o1 both have error rates of exactly .5 because the test sets consist of unseen bigrams, which are assigned a probability of 0 by the maximum likelihood estimate. Since we chose to form pseudo-words out of verbs of similar frequencies, the back-off models BO-1 and BO-o1 also perform poorly.

	T_1	T_2	T_3	T_4	T_5
MLE-1	.5	.5	.5	.5	.5
MLE-o1	″	″	″	″	″
BO-1	0.517	0.520	0.512	0.513	0.516
BO-o1	0.517	0.520	0.512	0.513	0.516

Table 4.4: Base language model error rates

Since the back-off models consistently performed worse than the MLE models, we chose to use only the MLE models in our subsequent experiments. Therefore, we only ran comparisons between the measures that could utilize unsmoothed data, namely, the L_1 norm, the total divergence to the mean, and the confusion probability. It should be noted, however, that on BO-1 data, the KL divergence performed slightly better than the L_1 norm; in the next chapter, we will study the performance of the KL divergence more carefully.

4.5.3 Sample Closest Words

In this section, we examine the closest words to a randomly selected noun, "guy", according to the three measures L_1, A, and P_C.

Table 4.5 shows the ten closest words, in order, when the base language model is MLE-1. There is some overlap between the closest words for L_1 and the closest words for A, but very little overlap between the closest words for these measures and the closest words with respect to P_C: only the words "man" and "lot" are common to all three. Also observe that the word "guy" itself is only fourth on the list of words with the highest confusion probability with respect to "guy".

Let us examine the case of the nouns "kid" and "role" more closely. According to the similarity functions L_1 and A, "kid" is the second closest word to "guy", and "role" is considered relatively distant. In the P_C case, however, "role" has the highest confusion probability with respect to "guy," whereas "kid" has only the 80th highest confusion probability. What accounts for the difference between A and L_1 on the one hand and P_C on the other?

Table 4.6, which gives the ten verbs most likely to occur with "guy", "kid", and "role", indicates that both L_1 and A rate words as similar if they tend to cooccur with the same words in $\mathcal{Y}$. Observe that four of the ten most likely verbs to occur with "kid" are also very likely to occur with "guy", whereas only the verb "play" commonly occurs with both "role" and "guy".

If we sort the verbs by decreasing $P(y|\text{"guy"})/P(y)$, a different order emerges (table 4.7): "play", the most likely verb to cooccur with "role", is ranked higher than "get", the

L		A		P_C	
GUY	0.000000	GUY	0.000000	role	0.032925
kid	1.229067	kid	0.304297	people	0.024149
lot	1.354890	thing	0.329062	fire	0.013092
thing	1.394644	lot	0.330871	GUY	0.012744
man	1.459825	man	0.350695	man	0.011985
doctor	1.460766	mother	0.368966	year	0.009801
girl	1.479976	doctor	0.369644	lot	0.009477
rest	1.485358	friend	0.372563	today	0.009095
son	1.497497	boy	0.373881	way	0.008778
bit	1.497502	son	0.375474	part	0.008772
(role: rank 173)		(role: rank 43)		(kid: rank 80)	

Table 4.5: 10 closest words to the word "guy" for A, L, and P_C, using MLE-1 as the base language model. The rank of the words "role" and "kid" are also shown if they are not among the top ten.

Object	Most Likely Verbs
guy	see get play let give catch tell do pick need
kid	**get see** take help want **tell** teach send **give** love
role	**play** take lead support assume star expand accept sing limit

Table 4.6: For each object x, the ten verbs y with highest $P(y|x)$. Boldface verbs occur with both the given noun and with "guy." The base language model is MLE-1.

most likely verb to cooccur with "kid", thus indicating why "role" has a higher confusion probability with respect to "guy" than "kid" does.

Finally, we examine the effect of deleting singletons from the base language model. Table 4.8 shows the ten closest words, in order, when the base language model is MLE-o1. The relative order of the four closest words remains the same; however, the next six words are quite different from those for MLE-1. This data suggests that the effect of singletons on calculations of similarity is quite strong, as is borne out by the experimental evaluations described in section 4.5.4. We conjecture that this effect is due to the fact that there are many very low frequency verbs y in the data. Omitting singletons involving such words could then drastically alter the number of y's that cooccur with both x and x'. Since our similarity functions depend on such words, it is perhaps not so surprising that the effect on

(1) electrocute (2) shortchange (3) bedevil (4) admire (5) bore (6) fool (7) bless $\cdots$ (26) play $\cdots$ (49) get $\cdots$

Table 4.7: Verbs with highest $P(y|\text{"guy"})/P(y)$ ratios. The numbers in parentheses are ranks.

L		A		P_C	
GUY	0.000000	GUY	0.000000	role	0.050326
kid	1.174243	kid	0.300681	people	0.024545
lot	1.395178	thing	0.321719	fire	0.021434
thing	1.407363	lot	0.346137	GUY	0.017669
reason	1.416542	mother	0.364610	work	0.015519
break	1.424242	answer	0.366333	man	0.012445
ball	1.438618	reason	0.367112	lot	0.011255
answer	1.440296	doctor	0.373428	job	0.010992
tape	1.448657	boost	0.377174	thing	0.010919
rest	1.452688	ball	0.381274	reporter	0.010551

Table 4.8: 10 closest words to the word "guy" for A, L, and P_C, using MLE-o1 as the base language model.

similarity values of deleting singletons is rather dramatic. In contrast, a back-off language model is not as sensitive to missing singletons because of the Good-Turing discounting of small counts and inflation of zero counts.

4.5.4 Performance of Similarity-Based Methods

Figure 4.2 shows the error rate results on the five test sets, using MLE-1 as the base language model. The parameter β was always set to the optimal value for the corresponding parameter training set. RAND, which is shown for comparison purposes, simply chooses the weights $W(x, x')$ randomly. $\mathcal{S}(x)$ was set equal to $\mathcal{X}$ in all cases.

The similarity-based methods consistently outperform the MLE method (which, recall, always had an error rate of .5) and Katz's back-off method (which always had an error rate of about .51) by a huge margin; therefore, we conclude that similarity information is very useful for unseen word pairs where unigram frequency is not informative. The similarity-based methods also do much better than RAND, which indicates that it is not enough to simply combine information from other words arbitrarily: it is quite important to take word similarity into account. In all cases, A edged out the other methods. The average improvement in using A instead of P_C is .0082; this difference is significant to the .1 level ($p < .085$) according to the paired t-test.

The results for the MLE-o1 case are depicted in figure 4.3. Again, we see the similarity-based methods achieving far lower error rates than the MLE, back-off, and RAND methods, and again, A always performed the best. However, omitting singletons amplified the disparity between A and P_C: the average difference in their error rates increases to .024, which is significant to the .01 level (paired t-test).

An important observation is that all methods, including RAND, were much more effective if singletons were included in the base language model; thus, in the case of unseen word pairs, it is clear that singletons should not be ignored by similarity-based models.

Recall that in these experiments we set $\mathcal{S}(x) = \mathcal{X}$. From the point of view of computational efficiency, it may not be desirable to sum over all the words in $\mathcal{X}$. We experimented with using only the k closest words to x, where k varied from 100 to 1000 ($= |\mathcal{X}|$). We see from figure 4.4 that stopping at $k = 600$ is sufficient to capture most of the performance

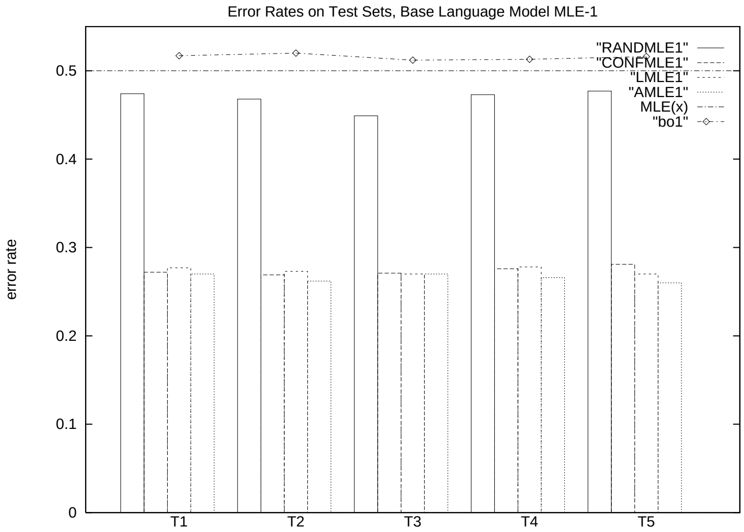

Figure 4.2: Error rates for each test set, where the base language model was MLE-1. The methods, going from left to right, are RAND , P_C, L_1, and A, and the performances shown are for settings of β that were optimal for the corresponding training set. β values for L_1 ranged from 4.0 to 4.5. β values for A ranged from 10 to 13.

improvement. It also appears that L_1 and A use the closest words more efficiently, as we could sum over 10 times fewer words ($k = 100$) at a performance penalty of less than 1%; stopping at $k = 100$ for P_C would result in increasing the error rate by 4%.

4.6 Conclusion

Automatically-derived similarity-based language models provide an appealing approach for dealing with data sparseness. We suggest a framework which relies on maximum likelihood estimates when reliable statistics are available, and uses similarity-based estimates only in situations where data is lacking.

We have described and compared the performance of four such models against two standard estimation methods, the MLE method and Katz's back-off scheme, on a pseudo-word disambiguation task. We observed that the similarity-based methods perform much better then the standard methods on unseen word pairs, with the method based on the KL divergence to the mean being the best overall.

We also investigated Katz's claim that one can build more compact language models without suffering significant performance degradation by discarding singletons in the training data. Our results indicate that for similarity-based language modeling, singletons are quite important; their omission leads to noticeably higher error rates.

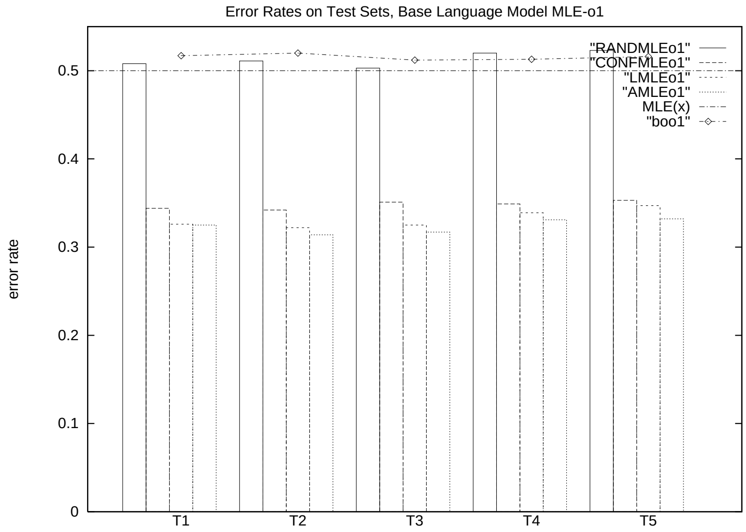

Figure 4.3: Error rates for each test set, where the base language model was MLE-o1. The methods, going from left to right, are RAND , P_C, L_1, and A, and the performances shown are for settings of β that were optimal for the corresponding training set. β values for L_1 ranged from 6 to 11. β values for A ranged from 21 to 22.

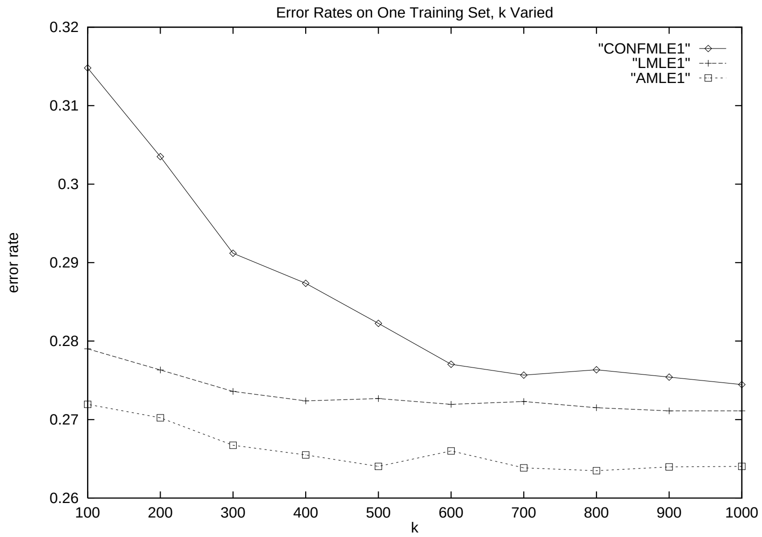

Figure 4.4: Error rates for one training set as k varies, where the base language model was MLE-1. β was set to the optimal value (4.5 for L, 13 for A).

Chapter 5

Similarity-Based Estimation for Speech Recognition

The previous chapter looked at the performance of three similarity-based methods on a simple disambiguation task. This chapter tackles the more realistic problems of perplexity reduction and speech-recognition error reduction. The distance function used here is the KL divergence, and the base language model is the back-off estimate. The similarity-based model considered in this chapter is based on the model developed in chapter 4, but has several added features meant to improve both performance and efficiency.

5.1 Introduction

Chapter 4 introduced similarity-based methods, developed a general framework for them, and compared several such methods on a pseudo-word disambiguation task. The pseudo-word task was very convenient from an experimental point of view; it allowed us to limit the number of senses a (pseudo-) word could have, as well as control the probabilities of the different senses (recall that we chose to create pseudo-words out of verbs with similar frequencies). Thus, we were able to perform a very clean experiment to demonstrate that indeed, similarity-based methods do have the potential to outperform standard approaches to sparse data problems.

However, it must be admitted that pseudo-word disambiguation seems a bit distant from problems encountered in real-word applications. Therefore, in this chapter we evaluate a similarity-based method on two tasks: perplexity reduction and speech-recognition error rate.

Perplexity is often used as a performance metric for language modeling systems; it is generally assumed that lowering the perplexity is correlated with better performance in practice (Jelinek, Mercer, and Roukos, 1992). Let P_{LM} be a probability model and S some[1] sample of text. Then the perplexity PP measures how well P_{LM} models S:

$$\mathrm{PP} = P_{LM}(S)^{-1/|S|}.$$

The intuition behind this expression is that a good language model should assign high

[1] Jelinek et al. note that the perplexity is a more accurate measure of the difficulty of recognition if the sample is large.

probability (and therefore low perplexity) to S, since S was generated by the (unknown) source distribution for the language. Another way to look at it is to regard the perplexity as measuring the average branching of the text from the point of the language model. For example, suppose we have two language models, P_1 and P_2. If it turns out that according to P_1, the only words that have a high probability of occurring after the word "San" are "Juan" and "Jose", whereas according to P_2, "Juan", "Jose", "cat", and "dog" all have a high probability of occurring after "San", then we would say that P_1 is the better language model. It is more certain about which words can follow "San"; one could say that it is less perplexed.

In this chapter, we model the probabilities of pairs of adjacent words rather than object-verb pairs; that is, $\mathcal{X} = \mathcal{Y}$, and the pair (x, y) refers to the event that the two-word sequence, or *bigram*, "$x\,y$" occurred in the training sample. We thus tackle the problem of *bigram language modeling*, which is a special case of *n-gram language modeling*; n-grams are the dominant language-modeling technology in speech recognition today. In a bigram language model, the probability of a string of words is factored into a product of conditional word pair probabilities:

$$P_{LM}(w_1 w_2 \ldots w_n) = \prod_i P_{LM}(w_i | w_{i-1}).$$

Then, the perplexity of a bigram model P_{LM} with respect to the string $w_1 w_2 \ldots w_n$ is

$$\left(\prod_i P_{LM}(w_i | w_{i-1}) \right)^{-1/n} = \exp \left(-\frac{1}{n} \sum_i \log P_{LM}(w_i | w_{i-1}) \right),$$

where base 10 logarithm and exponential functions are used throughout this chapter, as in chapter 4.

Given our concern with the practicality of similarity-based estimation, we will also consider several heuristics for improving the efficiency and performance of similarity-based models. In particular, we will be interested in the effect of limiting the number of similar words that are consulted in making an estimate for a particular bigram. Another heuristic we apply is to interpolate the similarity information with the *unigram* (single word) probability used by Katz's back-off method. We find that combining these two estimates does improve performance, although it is best not to rely too much on the unigram probability (this is a gratifying result, as it tells us that the similarity information is more important than the unigram information).

The rest of this chapter proceeds as follows. Section 5.2 explains the modifications we make to the similarity-based model introduced in the previous chapter. Section 5.3 presents our evaluation results: the new similarity model achieved a 20% reduction in perplexity with respect to Katz's back-off model on unseen bigrams in *Wall Street Journal* data. These constituted just 10.6% of the test sample, leading to an overall reduction in test-set perplexity of 2.4%. We also experimented with an application of our language modeling technique to speech recognition, and found that it yielded a statistically significant reduction in recognition error. Section 5.4 points out some directions for further research.

5.2 The Similarity Model

Recall the general form for similarity-based models developed in chapter 4:

$$\hat{P}(y|x) = \begin{cases} P_d(y|x) & \text{if } C(x,y) > 0 \\ \alpha(x)P_r(y|x) & \text{otherwise } ((x,y) \text{ is unseen}) \end{cases} \quad . \tag{2.3}$$

We defined P_r to be P_{SIM}, where

$$P_{SIM}(y|x) = \sum_{x' \in \mathcal{S}(x)} \frac{W(x,x')}{\sum_{x' \in \mathcal{S}(x)} W(x,x')} P(y|x'). \tag{4.2}$$

Now, in the last chapter, we simply set $\mathcal{S}(x)$, the set of objects most similar to x, to be equal to the set $\mathcal{X}$. From a computational standpoint, though, this is somewhat unsatisfactory if $\mathcal{X}$ is large. Furthermore, it might well be the case that only a few of the closest objects contribute to the sum in (4.2). Therefore, we experiment in this chapter with limiting the size of $\mathcal{S}(x)$. We now introduce parameters k and t, and define $\mathcal{S}(w_1)$ to be the set of at most k words w_1' (excluding w_1 itself) that satisfy $D(w_1 \parallel w_1') < t$. We need to tune k and t experimentally.

We will use the KL divergence as distance function in the experiments described below, since we did not provide performance results for it in the previous chapter. Recall that the weight function $W(x,x')$ for the KL divergence was defined to be

$$W(x,x') = 10^{-\beta D(x \| x')} \quad .$$

Again, the parameter β controls the relative contribution of words at different distances from x: as β increases, the nearest words to β get relatively more weight. As β decreases, remote words have a larger effect on the sum (4.2). Like k and t, β is tuned experimentally.

While in the preceding chapter we set P_r to be P_{SIM}, we shall see that it is better to smooth P_{SIM} by interpolating it with the unigram probability $P(y)$ (recall that Katz used $P(y)$ as $P_r(y|x)$). Using linear interpolation we get

$$P_r(y|x) = \gamma P(y) + (1-\gamma)P_{SIM}(y|x) \ , \tag{5.1}$$

where γ is an experimentally-determined interpolation parameter. This smoothing appears to compensate for inaccuracies in $P_{SIM}(y|x)$, mainly for infrequent conditioning words. However, as the evaluation below shows, good values for γ are small, that is, the similarity-based model plays a stronger role than the independence assumption.

To summarize, we construct a similarity-based model for $P(y|x)$ and then interpolate it with $P(y)$. The interpolated model (5.1) is used as the probability redistribution model P_r in (2.3) to obtain better estimates for unseen bigrams. Four parameters, to be tuned experimentally, are relevant for this process: k and t, which determine the set of similar words to be considered, β, which determines the relative effect of these words, and γ, which determines the overall importance of the similarity-based model.

k	t	β	γ	training reduction (%)	test reduction (%)
60	2.5	4	0.15	18.4	20.51
50	2.5	4	0.15	18.38	20.45
40	2.5	4	0.2	18.34	20.03
30	2.5	4	0.25	18.33	19.76
70	2.5	4	0.1	18.3	20.53
80	2.5	4.5	0.1	18.25	20.55
100	2.5	4.5	0.1	18.23	20.54
90	2.5	4.5	0.1	18.23	20.59
20	1.5	4	0.3	18.04	18.7
10	1.5	3.5	0.3	16.64	16.94

Table 5.1: Perplexity reduction on unseen bigrams for different model parameters

5.3 Evaluation

We evaluated our method by comparing its perplexity and effect on speech-recognition accuracy with the baseline bigram back-off model developed by MIT Lincoln Laboratories for the *Wall Street Journal* (WSJ) text and dictation corpora provided by ARPA's HLT program (Paul, 1991).[2] The baseline back-off model closely follows the Katz design discussed in section 2.2, except that for the sake of compactness all singleton bigrams are treated as unseen (recall that this omission of singletons was quite detrimental to the simple similarity-based models considered in the previous chapter). The counts used in this model and in ours were obtained from 40.5 million words of WSJ text from the years 1987-89.

For the perplexity evaluation, we tuned the similarity model parameters by minimizing perplexity via a simple grid search on an additional sample of 57.5 thousand words of WSJ text drawn from the ARPA HLT development test set. The best parameter values found were $k = 60$, $t = 2.5$, $\beta = 4$ and $\gamma = 0.15$. For these values, the improvement in perplexity for unseen bigrams in a held-out 18 thousand word sample, in which 10.6% of the bigrams are unseen, is just over 20%. This improvement on unseen bigrams corresponds to an overall test set perplexity improvement of 2.4% (from 237.4 to 231.7). Table 5.1 shows reductions in training and test perplexity, sorted by training reduction, for different choices of k. The values of β, γ and t are the best ones found for each k.

From equation (4.2), it is clear that the computational cost of applying the similarity model to an unseen bigram is $O(k)$. Therefore, lower values of k (and t as well) are computationally preferable. From the table, we can see that reducing k to 30 incurs a penalty of less than 1% in the perplexity improvement, so relatively low values of k appear to be sufficient to achieve most of the benefit of the similarity model. As the table also shows, the best value of γ increases as k decreases, that is, for lower k a greater weight is given to the conditioned word's frequency. This suggests that the predictive power of neighbors beyond the closest 30 or so can be modeled fairly well by the overall frequency of the conditioned word.

The bigram similarity model was also tested as a language model in speech recognition.

[2]The ARPA WSJ development corpora come in two versions, one with verbalized punctuation and the other without. We used the latter in all our experiments.

B	commitments ... from leaders **felt the** three point six billion dollars
S	commitments ... from leaders fell to three point six billion dollars
B	followed by France the US **agreed in** Italy
S	followed by France the US Greece ... Italy
B	he whispers to **made a**
S	he whispers to an aide
B	the necessity for change **exist**
S	the necessity for change exists
B	without ... additional reserves Centrust would have reported
S	without ... additional reserves **of** Centrust would have reported
B	in the darkness past the church
S	in the darkness **passed** the church

Table 5.2: Speech recognition disagreements between models

The test data for this experiment were pruned word lattices for 403 WSJ closed-vocabulary test sentences. Arc scores in those lattices are sums of an acoustic score (negative log likelihood) and a language-model score, in this case the negative log probability provided by the baseline bigram model.

From the given lattices, we constructed new lattices in which the arc scores were modified to use the similarity model instead of the baseline model. We compared the best sentence hypothesis in each original lattice and in the modified one, and counted the word disagreements in which one of the hypotheses is correct. There were a total of 96 such disagreements. The similarity model was correct in 64 cases, and the back-off model in 32. This advantage for the similarity model is statistically significant at the 0.01 level. The overall reduction in error rate is small (from 21.4% to 20.9%) because the number of disagreements is small compared with the overall number of errors in the recognition setup used in these experiments.

Table 5.2 shows some examples of speech recognition disagreements between the two models. The hypotheses are labeled 'B' for back-off and 'S' for similarity, and the bold-face words are errors. The similarity model seems to be better at modeling regularities such as semantic parallelism in lists and avoiding a past tense form after "to." On the other hand, the similarity model makes several mistakes in which a function word is inserted in a place where punctuation would be found in written text.

5.4 Further Research

The model presented in this chapter provides a modification of the scheme for similarity-based estimation described in the preceding chapter; several heuristics for improving speed and performance were incorporated. We have demonstrated that the augmented model can be of use in practical speech recognition systems. We now discuss some possible further directions to explore.

It may be possible to simplify the current model parameters somewhat, especially with respect to the parameters t and k used to select the nearest neighbors of a word. On the other hand, it may be the case that using the same t and k for all words is too simplistic, although training a model in which t and k differ from word to word would involve massive sparse data problems.

A more substantial variation would be to base the model on the similarity between conditioned words (y) rather than on the similarity between conditioning words (x). For example, Essen and Steinbiss's variation 1 considers the confusion probability (4.3) of contexts rather than objects (Essen and Steinbiss, 1992). However, they noted that model 1-A was equivalent to model 2-B (which we discussed in section 4.4.2; it uses the confusion probability of conditioning words), and that their other model using variation 1 did not perform as well.

Other evidence may be combined with the similarity-based estimate. For instance, it may be advantageous to weigh the similarity-based estimate by some measure of the reliability of the similarity function and of the neighbor distributions. A second possibility is to take into account negative evidence, as Hatzivassiloglou and McKeown (1993) did (see the discussion in section 3.6.1). For example, if x is frequent, but y never followed it, there may be enough statistical evidence to put an upper bound on the estimate of $P(y|x)$. This may require an adjustment of the similarity-based estimate, possibly along the lines of the work of Rosenfeld and Huang (1992).

Finally, the similarity-based model may be applied to configurations other than bigrams. For trigrams, it is necessary to measure similarity between different conditioning bigrams. This can be done directly, by measuring the distance between distributions of the form $P(w_3|w_1, w_2)$, corresponding to different bigrams (w_1, w_2). Alternatively, and more practically, it may be possible to define a similarity measure between trigrams as a function of the similarities between corresponding words in them.

5.5 Conclusions

Similarity-based models suggest an appealing approach to dealing with data sparseness. Based on corpus statistics, they provide analogies between words that often agree with our linguistic and domain intuitions. In the previous chapter we looked at the performance of various instantiations of a simple similarity-based model. In this chapter we presented a variant that provides noticeable improvement over Katz's back-off estimation method on realistic evaluation tasks.

The improvement we achieved for a bigram model is statistically significant, although it is modest in its overall effect because of the small proportion of unseen events. While we have used bigrams as an easily accessible platform to develop and test the model, more substantial improvements might be obtainable for more informative configurations. An obvious case is that of trigrams, for which the sparse data problem is much more severe. For example, Doug Paul (personal communication) reports that for WSJ trigrams over a 20000 word vocabulary, only 58.6% of the test set trigrams occurred in 40 million of words of training data.

Chapter 6

Conclusion

> This paper is an absolute leviathan! Reverberating with history and personal recollection and occasionally exploding with well-aimed critical bursts, it sweeps you up like a great tidal wave and carries you along for over one hundred pages at an accelerating tempo, leaving you at the end with a sense that its driving energy has still not spent itself.... (Rosenkrantz, 1983, pg. viii)

We have presented two ways to make use of distributional similarity for applications in natural language processing. The first was a distributional clustering method, which proved not only to create clusters that seem to correspond to intuitive sense distinctions but also to lead to a cluster-based language model with good predictive power. Our clustering method yields soft, hierarchical clusters. Our use of soft clustering in a language processing context appears to be novel, but is rather natural, since many words are ambiguous.

We also presented a nearest-neighbor approach, where we combined estimates from similar words rather than from cluster centroids. This approach has the advantage of computational efficiency, since we do not need to engage in the iterative estimation necessary in our clustering work. We showed that methods based on the KL divergence provided substantial improvement over Katz's back-off method for unseen word pairs, and noticeable improvement over Essen and Steinbiss's confusion probability on a pseudo-word disambiguation task. To further demonstrate that similarity information can be helpful for applications, we also showed that an extension of our similarity-based model can produce both perplexity reduction and speech recognition error-rate reduction.

We can only conclude that the incorporation of similarity information has the potential to provide better results in the area of language modeling. But our techniques may extend farther than that. Indeed, our clustering work certainly seems applicable to other problems such as automatic thesaurus construction or lexicon acquisition. The fact that the clusterings we produce are probabilistic may again be an advantage, since for instance words may appear in more than one thesaurus category. It would also be interesting to experiment with applying our techniques to the problems of document clustering and indexing, as mentioned at the end of chapter 3.

This brings up a deeper question, though. What is the proper way to evaluate the inherent quality of clusterings (as opposed to measuring the performance gain clusterings can provide)? We need a good way to talk about how different one clustering is from another in order to analyze competing clustering methods. As we move to larger and larger data sets, it becomes more and more impractical to perform an evaluation such as that described by Hatzivassiloglou and McKeown (1993) where automatically-derived classes were compared to classes created by humans. Perhaps one fruitful direction, at least for hierarchical clusterings, would be to look at edit distances between trees (see, e.g., Kannan, Warnow, and Yooseph (1995)).

Another key question to address is whether we can formulate *adaptive* versions of our algorithms. Since our methods do not rely on heavily annotated samples, it is easy to acquire new training data. What we would like is a way to incorporate new information without having to restart the clustering process (in the distributional clustering case) or recalculate the similarity matrix (in the nearest-neighbor case). For clustering, the fact that we use soft classes may once again provide the answer, since we can reestimate membership probabilities as new data comes in; if we built hard clusters, we would have to adjust for prematurely grouping two objects together or splitting two objects apart.

References

Abney, Steven. 1996. Statistical methods and linguistics. In Judith Klavans and Philip Resnik, editors, *The Balancing Act*. MIT Press, Cambridge, MA.

ACL. 1993. *31st Annual Meeting of the ACL*. Association for Computational Linguistics, Somerset, NJ. Distributed by Morgan Kaufmann, San Francisco, CA.

Aczél, Janos and Zoltán Daróczy. 1975. *On Measures of Information and Their Characterizations*, volume 115 of *Mathematics in Science and Engineering*. Academic Press, New York.

Anderberg, Michael R. 1973. *Cluster Analysis for Applications*, volume 19 of *Probability and Mathematical Statistics*. Academic Press, New York.

Bahl, Lalit R., Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190, March.

Bezdek, James C. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced Applications in Pattern Recognition. Plenum Press, New York, NY.

Black, Paul and Joseph Kruskal. 1997. Comparative lexicostatistics: A brief history and bibliography of key works. `http://www.ntu.edu.au/education/langs/ielex/-BIBLIOG.html`.

Brown, Peter F., Vincent J. DellaPietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.

Brucker, Peter. 1978. On the complexity of clustering problems. In Rudolf Henn, Bernhard H. Korte, and Werner Oettli, editors, *Optimization and Operations Research*, number 157 in Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin.

Cheeseman, Peter, James Kelly, Matthew Self, John Stutz, Will Taylor, and Don Freeman. 1988. AutoClass: A Bayesian classification system. In John Laird, editor, *Proceedings of the Fifth International Machine Learning Conference*, pages 54–64, Ann Arbor, MI, June. Morgan Kaufmann.

Cheeseman, Peter and John Stutz. 1996. Bayesian classification (AutoClass): Theory and results. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, Menlo Park, CA, pages 153–180.

Chen, Stanley F. and Joshua T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the ACL*, pages 310–318. ACL, Association for Computational Linguistics, Somerset, NJ. Distributed by Morgan Kaufmann, San Francisco, CA.

Chomsky, Noam. 1964. *Syntactic Structures*. Number IV in Janua Linguarum. Mouton, The Hague, The Netherlands.

Church, Kenneth. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143.

Church, Kenneth W. and William A. Gale. 1991. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, 5:19–54.

Church, Kenneth Ward and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March.

Classification Society of North America. 1996. Classification literature automated search service. `http://www.pitt.edu/~csna/class.html`.

Cover, Thomas M. and Joy A. Thomas. 1991. *Elements of Information Theory*. Wiley series in telecommunications. Wiley-Interscience, New York.

Cutting, Douglass R., David R. Karger, Jan O. Pedersen, and John W. Tukey. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. In *15th Annual International SIGIR*, pages 318–329, Denmark, June.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *35th Annual Meeting of the ACL*, pages 56–63. ACL, Association for Computational Linguistics, Somerset, NJ. Distributed by Morgan Kaufmann, San Francisco, CA.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1995. Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, 9:123–152.

Dagan, Ido, Fernando Pereira, and Lillian Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. In *32nd Annual Meeting of the ACL*, pages 272–278. ACL, Association for Computational Linguistics, Somerset, NJ. Distributed by Morgan Kaufmann, San Francisco, CA.

Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38.

Digital Equipment Corporation. 1997. Altavista search: Main page. `http://www.altavista.digital.com/`.

Essen, Ute and Volker Steinbiss. 1992. Co-occurrence smoothing for stochastic language modeling. In *Proceedings of ICASSP*, volume 1, pages 161–164.

Finch, Steven P. and Nicholas J. Chater. 1992. Bootstrapping syntactic categories. In *Proceedings of the 14th Annual Conference of the Cognitive Science Society of America*, pages 820–825.

Firth, John Rupert. 1957. A synopsis of linguistic theory 1930–1955. In Philological Society, editor, *Studies in Linguistic Analysis*. Blackwell, Oxford, pages 1–32. Reprinted in *Selected Papers of J. R. Firth*, edited by F. Palmer. Longman, 1968.

Fisher, Ronald A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.

Gale, William, Kenneth Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *Working Notes, AAAI Fall Symposium Series, Probabilistic Approaches to Natural Language*, pages 54–60.

Gibbons, Jean Dickinson. 1993. *Nonparametric Measures of Association*. Number 07-091 in Sage University Paper series on Quantitative Applications in the Social Sciences. Sage, Newberry Park, CA.

Good, Irving J. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3,4):237–264.

Green, Phil. 1996. Lecture notes for Genome Sequence Analysis (University of Washington class MBT 599C), fall 1996. `http://www.genome.washington.edu/-MBT599C/lecture10.edit.html`, October. Notes taken by David Adams. Also available at `http://www.eecs.harvard.edu/~llee/green.html`.

Grishman, Ralph and John Sterling. 1993. Smoothing of automatically generated selectional constraints. In *Human Language Technology*, pages 254–259, San Francisco, California. Advanced Research Projects Agency, Software and Intelligent Systems Technology Office, Morgan Kaufmann.

Hanson, Robin, John Stutz, and Peter Cheeseman. 1991. Bayesian classification with correlation and inheritance. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, volume 2, pages 692–698, San Mateo, August. International Joint Conferences on Artificial Intelligence, Morgan Kaufmann.

Hartigan, John A. 1975. *Clustering Algorithms*. Wiley series in probability and mathematical statistics. Wiley-Interscience, New York.

Hatzivassiloglou, Vasileios and Kathleen McKeown. 1993. Towards the automatic identification of adjectival scales: Clustering of adjectives according to meaning. In *31st Annual Meeting of the ACL*, pages 172–182, Columbus, OH.

Hindle, Don. 1990. Noun classification from predicate-argument structures. In *28th Annual Meeting of the ACL*, pages 268–275. ACL, Association for Computational Linguistics, Somerset, NJ. Distributed by Morgan Kaufmann, San Francisco, CA.

Hindle, Donald. 1994. A parser for text corpora. In B. T. Sue Atkins and Antonio Zampolli, editors, *Computational Approaches to the Lexicon*. Oxford University Press, Oxford, England, chapter 5, pages 103–151.

Hofmann, Thomas and Joachim M. Buhmann. 1997. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, January.

Jaynes, Edwin T. 1957. Information theory and statistical mechanics. *Physical Review*, 106:620–630.

Jaynes, Edwin T. 1983. Brandeis lectures. In Roger D. Rosenkrantz, editor, *E.T. Jaynes: Papers on Probability, Statistics, and Statistical Physics*, volume 158 of *Synthese Library*. D. Reidel, Dordrecht, Holland, chapter 4, pages 39–76. Lectures given at Brandeis University in 1962.

Jelinek, Frederick and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands: North-Holland, May.

Jelinek, Frederick, Robert L. Mercer, and Salim Roukos. 1992. Principles of lexical language modeling for speech recognition. In Sadaoki Furui and M. Mohan Sondhi, editors, *Advances in Speech Signal Processing*. Mercer Dekker, Inc., pages 651–699.

Jones, William P. and George W. Furnas. 1987. Pictures of relevance. *Journal of the American Society for Information Science*, 38(6):420–442, November.

Kannan, Sampath, Tandy Warnow, and Shibu Yooseph. 1995. Computing the local consensus of trees. In *Sixth ACM-SIAM Symposium on Discrete Algorithms*, pages 68–77.

Karov, Yael and Shimon Edelman. 1996. Learning similarity-based word sense disambiguation from sparse data. In *4rth Workshop on Very Large Corpora*, pages 42–55. Also available as CS-TR 96-05, The Weizmann Institute of Science, and cmp-lg/9605009.

Katz, Slava 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, March.

Kaufman, Leonard and Peter J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley series in probability and mathematical statistics. John Wiley and Sons, New York.

Kerridge, David F. 1961. Inaccuracy and inference. *Journal of the Royal Statistical Society, Series B*, 23:184–194.

Kneser, Reinhard and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modelling. In *European Conference on Speech Communications and Technology*, pages 973–976, Berlin, Germany.

Knuth, Donald E. 1973. *The Art of Computer Programming*, volume 1 (Fundamental Algorithms) of *Addison-Wesley Series in Computer Science and Information Processing*. Addison-Wesley, Reading, MA, second edition.

Kullback, Solomon. 1959. *Information Theory and Statistics*. John Wiley and Sons, New York.

Luk, Alpha K. 1995. Statistical sense disambiguation with relatively small corpora using dictionary definitions. In *33rd Annual Meeting of the ACL*, pages 181–188. ACL, Association for Computational Linguistics, Somerset, NJ. Distributed by Morgan Kaufmann, San Francisco, CA.

Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Nádas, Arthur. 1985. On Turing's formula for word probabilities. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-33(6):1414–1416, December.

Ney, Hermann and Ute Essen. 1993. Estimating 'small' probabilities by leaving-one-out. In *Third European Conference On Speech Communication and Technology*, pages 2239–2242, Berlin, Germany.

Paul, Douglas B. 1991. Experience with a stack decoder-based HMM CSR and back-off n-gram language models. In *Proceedings of the Speech and Natural Language Workshop*, pages 284–288, Palo Alto, California, February. Defense Advanced Research Projects Agency, Information Science and Technology Office, Morgan Kaufmann.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *31st Annual Meeting of the ACL* (ACL, 1993), pages 183–190.

Pinker, Steven. 1984. *Language Learnability and Language Development.* Number 7 in Cognitive science series. Harvard University Press, Cambridge, MA.

Rényi, Alfréd. 1970. *Probability Theory.* North-Holland, Amsterdam.

Resnik, Philip. 1992. Wordnet and distributional analysis: A class-based approach to lexical discovery. *AAAI Workshop on Statistically-based Natural Language Processing Techniques*, pages 56–64, July.

Resnik, Philip. 1993. Selection and information: A class-based approach to lexical relationships. IRCS Report 93-42, University of Pennsylvania, Philadelphia, PA, December. Author's Ph.D. thesis.

Rose, Kenneth, Eitan Gurewitz, and Geoffrey C. Fox. 1990. Statistical mechanics and phase transitions in clustering. *Physical Review Letters*, 65(8):945–948.

Rosenfeld, Ronald and Xuedong Huang. 1992. Improvements in stochastic language modeling. In *DARPA Speech and Natural Language Workshop*, pages 107–111, Harriman, New York, February. Morgan Kaufmann, San Mateo, California.

Rosenkrantz, Roger D. 1983. Preface. In Roger D. Rosenkrantz, editor, *E.T. Jaynes: Papers on Probability, Statistics, and Statistical Physics*, volume 158 of *Synthese Library.* D. Reidel, Dordrecht, Holland, pages vii–ix.

Ruspini, Enrique H. 1970. Numerical methods for fuzzy clustering. *Information Sciences*, 2:319–350.

Salton, Gerard. 1968. *Automatic Information Organization and Retrieval.* McGraw-Hill computer science series. McGraw-Hill, New York.

Schütze, Hinrich. 1992. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 113–120.

Schütze, Hinrich. 1993. Part-of-speech induction from scratch. In *31st Annual Meeting of the ACL* (ACL, 1993), pages 251–258.

Signal Processing Society of the IEEE. 1996. *ICASSP-96*, Atlanta, GA, May. IEEE, New York, NY.

Skilling, John. 1991. Fundamentals of MaxEnt in data analysis. In Brian Buck and Vincent A. Macaulay, editors, *Maximum Entropy in Action: A Collection of Expository Essays*. Oxford Univerity Press, Oxford, chapter 2, pages 19–40.

Sugawara, K., M. Nishimura, K. Toshioka, M. Okochi, and T. Kaneko. 1985. Isolated word recognition using hidden Markov models. In *Proceedings of ICASSP*, pages 1–4, Tampa, Florida. IEEE.

Ueberla, Joerg P. 1994. An extended clustering algorithm for statistical language models. Technical Report DRA/CIS(CSE1)/RN94/13, Forum Technology – DRA Malvern, December. Also available at `http://xxx.lanl.gov/ps/cmp-lg/9412003`.

Wallace, Christopher S. and David L. Dowe. 1994. Intrinsic classification by MML – the Snob program. In Chenqi Zhang, John Debenham, and Dickson Lukose, editors, *AI '94 – Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, pages 37–44, Armidale, NSW, Australia, November. World Scientific.

Yahoo! Inc. 1997. Yahoo! `http://www.yahoo.com/`.

Yarowsky, David. 1992a. CONC: Tools for text corpora. Technical Memorandum 11222-921222-29, AT&T Bell Laboratories.

Yarowsky, David. 1992b. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *COLING-92*, pages 454–460, August.

Zadeh, Lofti A. 1965. Fuzzy sets. *Information and Control*, 8:338–353.