Contents

Abstract

I describe a model of word usage that categorizes the various paradigmatic syntactic contexts in which words appear, without resort to traditional grammatical classifications or part of speech labels. The model is derived by inductively classifying the contexts of words as they appear in corpus, where contexts are compared according to structural similarities. The model is motivated by a desire to preserve the structural syntactic information which constrains word use, and to represent it in a useful and applicable form. As syntactic relations are constrained by word meaning, this also allows corpus driven induction of lexical semantic information.

A certain type of interword dependency can be described in terms of the relations between the context classes in which those words appear, much as is done traditionally by assigning part of speech labels before further analysis. The overall use of a word can then be described in terms of a set of substitution classes in which the word is assigned graded membership, according to its appearance in the context classes. As license to appear in a given syntactic situation is generally taken as a property of lexical semantics, each context class represents the confluence of semantic properties enabling its occurrence. By cataloging the major patterns of use for a word, and capturing those distinctions of paradigmatic syntactic behavior in which that word participates, we indicate the semantic features which that word must carry.

Because the space of possible context events is so much greater than the space of

individual word events, data reduction and modelling become an important consideration.

I also investigate here a method of parameterizing the entire space of possible syntactic contexts in a real-valued multi-dimensional space, rather than as a set of nominal features. In this representation, the use of a word is given by the distribution of its observed use across this space. By altering the focus of the representation from the aggregate use of a particular word to the relation among the possible uses of that word, this model provides a more basic representation, designed to represent the diverse and polymorphic behavior of words. The model differs from most existing statistical language models in that analogy is not based on a direct comparison among elements, but on a comparison in the smaller parameter space.

[already said this. remove it and put it litany of advantages.]

The model is derived by categorizing word contexts using an inductive clustering scheme. Various possible metrics for clustering, as well as various interpretations of these results will be discussed. The categories formed by this process provide two natural interpretations on the text: 1) a view of the context classes as analogous to part of speech or syntactic classes, where each class has a particular distribution in terms of its realization as words of the language; and 2) a view of the distribution of any given word across these categories, as a separation of the usage of that word into usage classes, or modes of use, which may be indicative of semantic properties.

Chapter 1

Introduction

The traditional goal of descriptive linguistics is to provide a model of the sequences used in a language. Any such model should satisfy two distinct goals: it should accurately model existing data as well as novel data generated by language users, and it should provide a compact, analytic representation for the data. Both of these goals require that the model abstract away from the specifics of observed data, in order to include possible utterances that have never occurred before, and to reduce the representation to a comprehensible size and form.

This abstraction usually takes the form of a categorization of words and word properties, a categorization of sequences of words into syntactic constituents, and a set of rules or constraints for combining categories into sequences in the language.

In this dissertation, I describe a model of word usage that categorizes the various paradigmatic syntactic contexts in which words appear, without resort to traditional grammatical classifications or part of speech labels. Words can then be described in terms of the range and distribution of context categories in which they are observed to occur. This type of categorial representation satisfies both the predictive capacity and compact representation goals of a useful model.

I show that such a model can be acquired by inductively classifying contexts as

they appear in corpus, where contexts are compared according to structural similarities. This model is *associative* in that it describes the use of a word or category through its typical associations with partial seqeunces of the language. These partial sequences, or pardigmatic contexts, are themselves expressed in terms of words or word categories.

1.1 Machine learning of Language Models

All techniques to model language rely on the presumption of *regular structure* in the language; that is, they assume that words and phrases are used with some regular correspondence to one another and to the situations in which they are used. If this regularity in the syntactic and semantic structure of language is truly present, we should be able to find it directly in the observable artifacts of language — in the spoken sounds or written words we use to communicate.

One of the main difficulties in finding these regularities, however, is the nature of the problem data itself. While linguists, being natural language users, are able to study both sides of the problem, the symbols along with their meaning, the theories and data sets have grown to such a size that many enlist the aid of computers to help with the task. The machines, however, are only able (at this point) to directly view one side of the problem, the symbols. Other information, such as part-of-speech, syntactic structure labeling, and semantic labeling, can be hand-encoded for the computer, but only at great cost to the researcher ([Johansson and Stenström, 1991, DeMarcken, 1990]). One must also be concerned about the consistency of such informant-supplied mark-up, with respect to disagreement between annotators, as well as within the work of any single annotator. New methods and new characterizations that are capable of describing language without incorporating interpretation data, but which still leave the researcher avenues for meaningful interpretation will have to emerge to make this new type of language study practicable.

An immense amount of language data exists in a computer accessible form today. If one wants to use this data to inform the understanding of language, however, one must have a clear model on which to base one's study. Traditional rule-based grammars can be induced from corpus data (e.g. [Fu and Booth, 1975]), but if the researcher does not supply a 'backbone' of high level rules, current techniques must either posit rules in some highly constrained form (CNF, for example), or else suffer through searching an enormous combinatorial space of rules. If one is applying induction techniques in order to investigate the underlying structure of the language, such a constrained normal form or a limited initial grammar may prevent discovery of interesting phenomena.

The difficulty of matching a strict rule-based grammar to the exigencies of real data, as found in speech recordings or newswire corpora, has lead to the extension of these grammars in terms of rule probabilities ([Black *et al.*, 1992, Ayuso *et al.*, 1992, ?]). The size and complexity of the training data to be modelled leads to grammars with enormous numbers of possible rules. By assigning chances to each of the various transitions from a given non-terminal symbol, these probabilistic grammars provide a mechanism for evaluating which possible combination of rules provides the best analysis for a given sentence. The training data supplied, however, must be marked with the syntactic categories and parse structure that one desires to learn. The analysis of the training data must be supplied by human interpretation; the procedure will only learn a compact representation for the supplied analysis of the data.

Unsupervised methods, which do not require pre-marked data, have also been applied to language modelling. Chief among those used today are Markov models, commonly known as 'n-gram' models ([Baker, 1975, ?]). These models use probablities computed from the training data to estimate the chances that a word or sound follows some already observed sequence. This is particularly useful in deciphering noisy sound input, where the sounds heard through a microphone may not exactly correspond

to the signals used in training — if one has already heard a string a words, the estimated probabilities can help limit the interpretation of the next sound in the sequence to the correct word. This representation does not lend itself to much deep or useful interpretation, however. The model, in the end, consists of a set of contingency tables, giving the chances any particular word occurring, for each possible preceeding sequence of $n-1$ words.

Markov models can also be used with labelled data. One area in which they excel is in learning part of speech tagging models ([?, ?]). Here, the sequence of words is used to predict the part of speech category of the following word. Many of the approaches to this task technically use *hidden* Markov models – the underlying model is a representation of the probability a given category will follow a sequence of other categories, but the surface data, the words to be labelled, provide only indirect evidence of the categories. This application has much greater chances for accuracy than do the full language models, because of the limitted number of parts of speech, and hence the limitted number of possible part of speech combinations.

1.1.1 Sparse data and solutions

An exact model of word use in context can only apply if lexical items are observed in precisely the same lexical environment. Of course, it rarely happens that one environment is ever precisely the same as one seen before. The chances are very slim that a sizeable lexical context will ever recur. Unless one is willing to forego exact correspondence within the context, or to use very small contexts (on the order of one or two lexical items), there is almost no chance that enough data will be available to build a complete or accurate model of use.

Most current advances in descriptive language modelling, such as the Markov models already discussed, take primarily the latter approach, and use very short contexts to describe the typical environments in which words appear. However, in for example a trigram model (in which the occurrence of two lexical items is used to predict a third), most possible combinations of three lexemes do not appear in even the largest collections of data, and further, most of the observed events occur only once. Worse yet, the distribution of words in the language is such that unseen trigrams are nevertheless likely to appear in any new data under analysis. Thus, models which exactly fit collected data will contain incorrect predictions about real and expected events.

Two major approaches have been developed for dealing with these problems of estimation and prediction of cooccurrences from sparse corpus data: classification and smoothing methods. Smoothing techniques seek to estimate the probabilities of cooccurrence events based on assumptions of the frequency distributions of event types and of individual words [Good, 1953, Katz, 1987]). Direct associations between the predictor words and the outcome are taken as too numerous to model individually, and so the interactions are predicted based in terms of the frequency classes of the component words. Thus, unseen events are accorded a probability modelled after the distribution of single words. This is an unfortunate result, since words have associations based on much more than their relative frequencies.

Classification methods

Class-based methods overcome some of the difficulties by modelling the interdependence of *classes* of 'similar' words [Brown *et al.*, 1992b, ?, ?]. Because of the relatively low number of classes and the correspondingly much lower number of parameters, modelling the conditional probabilities between classes is far more tractable than those between individual words. The lower number of parameters also provides a great deal of smoothing in the predicted probabilities, because much of the low-frequency sampling noise that affects word-based Markov models is eliminated due to the higher class frequencies. becomes tractable. Hoever, it may not be that all words

assigned to a class share the same associations, and that grouping them this way loses a great deal of the idiosyncratic behavior of each word.

Other methods form class-based representations by considering the 'average behavior' of words, using a composite representation made up of all the contexts in which a word has appeared [Finch and Chater, 1992, Schütze, 1993, Hughs, 1994]. By comparing the frequency of events within these aggregate representations, clusters of words with similar overall behavior can be found.

Although some of these methods claim to form representations of the syntactic behavior of the words, they fail to recognize that individual words may operate in a variety of functions, because they combine all observed contexts for a word into a single, averaged representation. Since all of these methods treat each word as an elemental unit and assign it to only one class, the rich polymorphic nature of word use is lost. An English word which exhibits a noun-verb alternations, say, is treated as a single type. While it may end up being classed with other words that have a similar alternation, associations between it and other words may have a strong dependency on the choice of nominal or verbal use, a distinction which is completely ignored by the model.

Recent similarity-based methods estimate the probabilities of collocations by using models of word similarity. Thus, the probability of a cooccurrence is modelled by analogy with events which have similar elements [?, Dagan *et al.*, 1994]. These have much in common with the class methods, but use sophisticated models of interdependence between the component words. Instead of using membership in a globally defined class, these methods use the set of words *most similar to* the target word as an analog for the target in order to judge a given cooccurrence. This set of similarities is of course different every word. Similarity between two words is measured by comparing the mutual information between each word in the pair and all other words in the lexicon. Words are considered similar to the extent that their profiles of mutual information match across the lexicon.

1.2 The associative model

The goal of this work is to build a structural model of word use. That is, to describe the use of each word in the language in terms of the structures it can appear with; and conversely, to describe each possible structure of the language in terms of the words allowed to participate in them. The model is categorical — a 'structure' in a language, in this sense, is a class of possible sequences of lexical items (words), differentiated from all other possible sequences. A lexical class, or syntactic category, is a set of words which may participate in a given structure. In the model I present, membership in either type of category can be *graded* – a particular sequence may partially belong to two or more structural categories, and a given word may be allowed to participate in two or more structures.

[short description of model]

The work presented here also represents a modeling technique which associates words based on the similarity of local cooccurrence patterns. As in the class-based Markov models, the immediate neighbors of a position are taken as predictors of the lexical item to be found in that position. However, instead of classifying the individual nieghbors of the target position, the entire neighborhood, the *context*, of the target item, is taken as a whole. This is accomplished using a similarity measure, based on the *edit distance* [Levenshtein,], that compares neighborhoods through the overlap of the contained lexical items and the similarity of their configuration.

Individual uses of lexical items can then be compared using the similarities of their contexts as an intermediate. Since we take the context as a syntactic predictor of the function of the target item, similarity between contexts implies similarity between the items found within them. This analogy process also allows us to separate the various

functions, or syntactic modes, a single item might exhibit. By being able to analyze each use of the item separately and independently, we gain a greater understanding of the variety of function is may take within the various contexts. Not only is this a finer-grain analysis than treating the word as a member of a class which has only a single descriptor, it goes below the level of word to elicit the variety of lexical behaviors for each spelling form. In theory, this decomposition has always been possible using n-gram Markov models, since each predictor sequence potentially indicates a different form of use of the target item, but the vast number of possible predictor sequences, and the lack of a relation among them, limits the practicality of such an approach. By introducing a similarity over the predictive environments, we enable this sort of lexical decomposition. By altering the focus of the representation from the aggregate use of a particular word to the relation among the possible uses of that word, this more basic representation is designed to represent the polymorphic nature of word use.

[diagram of focus word in context]

This type of model incorporates attributes of both the structural grammar descriptions and the sequential Markov models. It preserves the notion of grammatical structure, which captures the intuition that many separate strings of words can create the same functional grammatical construction. It also retains the explicit aspects of sequence that Markov models represent, i.e. one specific sequence can predict one specific outcome.

1.3 Outline of the text

Chapter 2 contains a discussion of the underpinnings of the associative model, beginning with an efficiency argument exploring the benefits gained through the use of syntactic structure. This discussion sets out a small number of basic syntactic structuring mechanisms that we might expect to find in language. The remainder of the chapter explains how we can use this knowledge to acquire a model of a language, and how the study of the purely formal (syntactic) structure of the language can lead to a better understanding of the semantic nature of the language system.

Chapter 3 outlines the specific formal qualities of the associative model, influenced by the nature of the available data and the specific goal of integrating formal with semantic processing. A further investigation into the qualities of the data leads directly to the definition of the lexical context and its descriptive properties. The class of associative models is described in terms of similarity measures defined by comparisons on these contexts. A discussion of the nature of sparse data leads to an explanation of the utility of classifications, both for syntactic generalizations and for word classes. The similarity measures are then expanded to include comparisons over classified data. The smoothing properties of the classification model are explored. The chapter completes with a comparison to other work on inductive classification of lexical data.

Chapter 4 discusses some appropriate context similarity measures which can be used as the basis for associative lexical models. These measures include edit distance, the localized edit distance, and set comparison measures. Differences in the measure which can affect the nature of the resulting model are discussed. Some experiments are presented which quantify the distinctions in the measures.

Other experiments are presented which give a baseline for these similarity measures, derived by assuming random distributional properties of the underlying language data. This baseline is compared to the distributions derived from a representative sample of English, and the differences are discussed.

Chapter 5 presents a number of techniques for deriving classifications based on the similarity measures. The properties of the classification techniques are explained in terms of their interactions with the similarity measures and model formation.

Experiments are presented which exhibit the agreement of the similarity measures and classifications with traditional syntactic analyses. An experiment demonstrating the stability and convergence of the classification techniques is also presented.

Chapter ?? describes how a model, once built for a particular data set, can be used in the analysis of additional data. Variaous visual analysis techniques are described in conjunction with this analysis.

Chapter ?? presents some experiments in building various associative lexical models. Included are some experiments on word sense identification.

Chapter 2

Language structure

This chapter outlines a very general theory of the constraints on language structure, as imposed by the efficiency of its use as a device for representation and communication. The immediate result, which should not be surprising, is that a small number of structural constraints, grammatical relations used in the representation of semantic relations, greatly improves the communicative efficiency of sequences of words, in terms of the number of words required and the processing power necessary to decode them.

The deeper result is that regularities in semantics should coincide with structural, syntactic regularities. Thus, any procedure which sets out to find certain structural, distributional regularities in the language will also find parallel semantic properties. This implies that a properly structured acquisition of distibutional properties can not only be used to form a syntactic model of a language, but can also be used to bootstrap a study of lexical semantics, and can form a basis for semantic inference procedures.

Chapter 3 provides methods for organising this type of lexical information. The following chapter (ch. 3) starts from the assumption that these structural mechanisms will be present in a language, and uses this information to organize a model of word use

within the language. The effectiveness of these and similar methods in the literature is purely contingent on the presence of these syntactic constraints.

[outline wrong – correct to match text]

The first sections (2.1) outlines what I mean by 'constraints' using familiar types of examples. The structural constraints discussed will be familiar to any student of syntax, particularly those familiar with the structuralist school, and also readers familiar with any variety of dependency grammar, including case grammars. While the structuralists describe these constraints in purely formal terms, based on their observed presence in languages, I argue that they are a necessary mechanism for communication.

The second section presents a number of familiar structural constraints, and gives an argument detailing their effects on representational efficiency. Although it is not an exhaustive description, I hope in this section to make clear at least the nature of the dependency between syntactic and semantic properties of words.

The later sections foreshadow the conclusions of the thesis, and hint at how these regularities might be exploited in order to derive conclusions about the semantic properties of words from observations of their use.

2.1 Constraints

The nature of the interdependence of words in a language, the *constraints* between words that govern their appearance with one another, are generally thought of as existing along a spectrum, running from purely formal, *syntactic* dependencies, to conceptual, *semantic* relations.

These constraints govern[1] the use and composition of utterances in that language,

[1] If one starts from a distributionalist perspective, this 'governance' is simply a post hoc expectation of what one will find in a sample of utterances from the language. If one views this as a cognitive model, however, one can talk of constraints controlling the actual construction of utterances by a

and describe the necessary conditions which ensure that a statement will be readily interpretable and understandable. The purely formal, syntactic, properties govern the configuration of words, constraining relative word order, without, some would say, any relation to the eventual meaning of the utterance. At the other extreme, semantic constraints which ensure that only words which construct meaningful relations are used together, ensuring that well-formed statements in the language are understandable.

That these two extremes are to some degree separable is evidenced by examples such as Chomsky's oft-quoted example:

Colorless green ideas sleep furiously. (2.1)

This is a statement which seems to have syntactic coherence, while giving no clear or ready interpretation. Chomsky's original argument [Chomsky, 1957] identifies this as "nonsensical", but "grammatical" (however, for competing arguments as to its sensibility, see [?]).

All of the words involved in this example are familiar to us; all of them have well known and separate meanings; yet, as a group, they have at least no clear or obvious meaning – especially in the sense that one would be hard-pressed to claim that their meanings are the same as in more 'normal' uses of these words. In fact, no adjacent pair of these words has any meaning together: ideas don't sleep, neither are they green, green things can't be colorless, and so on. However, the construction has the curious property that it sounds *almost* good, as if something could be adjusted slightly and the whole thing would fall into place.

I see these examples simply as defining the *types* of constraints allowable in language processing. Especially in subtle cases, the number of examples needed, pro and con, to demonstrate the existence, much less the the universality, of some particular syntactic constraint can be vary large, the boundary shifting this way and that as

speaker, and their interpretation by a listener. Either view will suffice for this discussion.

new and more clever examples are constructed. What is lacking, however, is a globally applicable analysis of how these constraints interact with words. For any particular example, an analysis can be constructed in one's favorite formalism to explain the interactions of the words, but a lexicon which outlines anything but a minute set of terms and their interdependent behavior is hard to come by.

[quote maybe Bran & James introduction on the teeny size of real lexicions. Explain that acquisition according to type, where the real nature of the type may be unknown, is preferable, in terms of effort expended, to postulation of type first, and then acquisition by known interaction.]

I will argue that classes of terms, all exhibitting the same constraints on use, can be acquired without defining those constraints or interactions, and without regard for syntactic formalism. Simply by taking the relational constraints of syntactic structure as a fundamental property of language, we are lead to the argument that observed parallels in usage may imply underlying parallels in meaning. By classifying utterances according to observed structural similarities, we can discover the distribution and interaction of those underlying causes within the lexicon. Because we can define degrees of structural similarity without having previously classified words according to possible meanings (such as part of speech labels, or more definite semantic types), we can define a set of types, and the constraints governing their interactions, free from intuitionist influences.

2.1.1 The range of effects

Example (2.1) should not convince one, however, that any string of words can be tossed together to have this property of linguistic intermediacy. Certainly, (2.1) is somehow much closer to being a real and meaningful expression in language than the following type of 'word salad':

eat direct Reluctantly have (fish ? mechanism the strengthen; (2.2)

The kinds of constraints evident in these two examples are certainly different in magnitude, but are they really different in nature? It is true that we must examine one much more closely than the other to realize that something is wrong with what we are reading, that the words on the page don't come together to make sense; but still the question remains, is there anything intrinsically different in what is wrong with the two? How can we characterize the differences in the way these examples fail to meet our expectations about what a well-constructed, intelligible sentence or utterance in a language should be?

Most would say that (2.2) is unacceptable because it does not meet the obvious formal[2] surface properties that are required of meaningful sentences — it does not match the *syntax* of English — not because of any deeper semantic considerations of failure in meaning. This sequence violates the normal arrangement of words, where 'normal' is somehow judged with regard to properties unrelated to specific meaning. If a word refers to something that is or could be participating in a relationship, we call it a *noun*. We know that, typically, in English, nouns appear before, but sometimes also after, a *verb*, which identifies the relationship, which is possibly one of action, between those nouns or to the world at large. *Quantifiers* appear before nouns, telling us to how many and to which items they might refer. These rules of placement and combination, which operate without regard to the deeper meaning of the words, are what is usually referred to as syntax or grammar.

One needn't look closely into the meaning of words in (2.2) to see that the verbs don't have nouns as subjects, the determiner is preceded by a noun and followed by a verb, and a host of other obvious form violations have been committed. It fails, by this

[2] 'Formal' here meaning 'to do with form or shape,' not necessarily implying any degree of precision or theoretical acceptance.

argument, because there are rules of English grammar that allow certain combinations of word types and disallow others.

Does the same sort of problem affect (2.1)? The more obvious surface rules seem to be followed explicitly here. There is a simple intransitive construction, 'ideas sleep'. The subject noun is modified by an adjective, and the resulting noun phrase has a further adjective modifier. The predicate is modified by an adverb. Even though this analysis doesn't help us make sense of the statement, everything is well positioned enough that we could, and many have, make a complete constituent analysis and draw a syntactic derivation tree for it. The meaning of such a tree would be questionable, though, since we are unsure whether the underlying meaningless sentence should really be allowed membership in the language.

We can decipher the example to this degree, however, only because of the relative familiarity of the words used and the roles they are used in. Fortunately, most words have a relatively fixed meaning, and appear most often in only one of these high-level grammatical role categories, making the listener's job of identifying the intended role of each word easier. When the rules of placement and combination are broken, however, the usual roles of the words come into conflict with the roles they must have in order to function meaningfully in combination.

The words used in (2.2) do not have nearly the flexibility in roles that they need in order to make the sequence an interpretable English sentence.

[need good role ambiguity example, not just attachment ambiguity]

Sources of information

The information deciding the role each word plays in a construction comes from at least two sources. One is lexical knowledge – knowledge of the word itself: what roles it usually takes in certain circumstances, what other terms it usually interacts with, and other specific knowledge of the one particular word. The other source is syntactic knowledge – our general knowledge of how words combine, applied to the environment in which the word is used. Information about the possible roles and meanings a word might assume is given to us by the arrangement of its neighbors. General knowledge of word arrangements limits serious problems, such as those of (2.2). More detailed information is also available — because a word must interact with its neighbors, our knowledge of how the neighbors normally operate imposes certain choices on how the word in focus can behave.

To illustrate the differences of these knowledge sources, we can construct an example with a novel word, with which we have no pre-associated lexical constraints. We don't know what this word might mean, but imagine it is used as in:

You should add some gardle to the soup. (2.3)

If one assumes that this sentence is interpretable, then we now know many things about the new word, all projected onto it by the remaining words of the construction. "add some X to the soup" tells us generally that *gardle* is a *thing*, a noun, since it is something you add (the lack of a right-hand nominal dependent rules out the possibility that it is a modifier of any sort; as in "some gardle turnips"). More specifically, it refers to a kind of thing that is composite and divisible, since it is "*some* gardle" rather than 'a' or 'the'. Also, it refers to something that goes into soup, and so is probably edible (that Scandinavian parsley people have been talking about, perhaps). The formal properties of syntactic combination have taken effect here, projecting an interpretation on the word, which seems acceptable because we have no prior knowledge about the word.

What would happen, though, if we were to take a familiar word and place it in the same construction? Let me use the word *strengthen*, which was so out of place in (2.2). We now have

You should add some strengthen to the soup. (2.4)

which seems totally incongruous, at least as much, if not more that it did in its appearance in (2.2). Our familiar interpretation of *strengthen* as a verb indicating an action clashes with every constraint on meaning that the remainder of the construction projects. No previous knowledge of this word prepares us to make such a leap in altering its familiar interpretation. The general syntax forces a nominal interpretation; the use of 'some' gives a composite reading; the verb, 'add', tells us the referent is manipulable; and the fact that it is added 'to the soup' gives us edibility, all of which conflict directly with its familiar interpretation.

2.1.2 Degree of constraint

[but I don't say much about degree; only about type.]

This example (2.4) sets up a complex matrix of projected constraints, all of which, from the nominal interpretation to the expectation of edibility, are broken by our existing knowledge of the word *strengthen*. All of the expectations that we have associated with the word, and all of the knowledge of its use in normal constructions, clashes with the relations forced upon it by the remainder of the construction.

The earlier *gardle* example (2.3), in contrast, is acceptable simply because the novel word does not carry any constraints along with it. It is merely acting as a carrier for the matrix of constraints projected upon it. We could use this word in place of practically *any* word, in any context, with identical results. While it may not have a predetermined meaning, we are more or less happy to allow its interpretation according to the constraints imposed by context. Further, those constraints tell us, more or less, what the word must mean.

Example (2.2) violates the primary syntactic constraints of the language in that even the most general level of configuration of the terms does not allow the application of the normal constraints between them. Each word is used in a context so foreign to its normal modes of use that we can not even identify any familiar pattern.

The 'green ideas' example (2.1) satisfies the basic agreements of context, but associations implied by the configuration fail to hold at a finer level of detail. In example (2.3), the implications of the configuration are quite acceptable, even though the word *gardle* contributes no information to the meaning. The final example, (2.4), contains a very different sort of error, one which the configuration imposes a coherent set of constraints, but that set clashes in every way with the familiar use of the offending word.

Syntactic constraints are related to word meaning

These examples demonstrate clearly that words place constraints on the words and constructions appearing jointly with them. *Any* configuration of words creates such constraints, whether they are in agreement or not. If one is to call the construction part of the language, however, the words associated under that configuration must not carry conflicting information. I take this as a fundamental observation about the structure of language. In this informational constraint view of language, effects that often are treated separately as syntactic or semantic are inextricably tied together — the syntactic configurations in which a word is used are dictated, in large part, by what that word means.

2.2 Structural origins

We have seen that there are definite limits on the manner in which words may combine in language. Is there anything in the concept of *language* itself that inherently limits the ways in which symbols might be put together to create complex relationships? Why is it that there should be syntactic structure at all? It is easy to form a group of words, simply tossed together like those of (2.2), for which there is no available interpretation. What is the structure of well-formed strings such that they form a

useful part of the language?

One problem quickly encountered persuing these questions is that we all have strong preconceptions of what 'language' is. For the present discussion, I take a rather loose view that language is primarily a representational system, using linear sequences of symbols to refer to mental constructs, which can exist in familiar predicate-argument relationships. The mental referents could be primitives, or higher-level composite objects. I would admit Sowa's conceptual graphs [Sowa, 1984], or other equivalent logic system, such frames or DRT [Kamp, 1981], as prototypical representations of the complex, multi-relational mental constructions. Language, as used here, is a system for relating linear strings of symbols to these highly structured, non-linear mental constructions.

How such a mapping is accomplished, and how that mapping results in constraints on allowable word combinations, are the topics of the following section. There is nothing, I believe, in the *functional* concept of language as a communicative and representational device that limits the ordering and structure of combinations. The conclusion supported here, however, is that it is the *practical*, *operational* concerns of instantiating and using a language that lead to configurational constraints on word use.

While a language system unconstrained in any way is conceptually feasible, it is not without implications. Some simple structural limitations greatly increase the functionality and economy of the system. Some of these economies, such as limitted memory capacity and sensory bandwidth, are inherently constrained by our neurological construction. We can begin by taking these as a practical starting point.

2.2.1 An unrestricted language

To begin to see where some structural constraints might arise, let us start by looking at the extreme alternative: a language with no syntactic properties at all. Imagine there exists a language $\mathcal{L}_0$ in which symbol order and configuration play no part at all in determining the functionality of the language, where I will define *functionality* as being the representational and communicative aspects of language, and which we can loosely refer to as the 'meaning' of statements.

For the present discussion, we can, without loss of generality, reduce the domain of the language to a simple 'blocks world' universe. The primitives represented in this universe are a known set of objects, physical (spatial) relations between those objects, and observable unary properties, such as color or shape.

Using no constraints of configuration, if a, b, and c are symbols in $\mathcal{L}_0$ each representing some primitive we state by assumption that the following sentences will have precisely identical interpretations:

$$\begin{array}{l} a\ b\ c \\ a\ c\ b \\ b\ a\ c \end{array} \tag{2.5}$$

To ground this, say, for instance, that a and b refer to objects in the world, two blocks, say, (but more generally any objects, physical or not – e.g. ideas, a chair, the observable property of blueness, etc.) and c is some symmetric relation that can exist between two things, such as being near. We could then paraphrase any of the expressions in (2.5) as "the things referred to by a and b are near each other." The non-configurational assumption allows us to re-order any string and claim that it must retain its interpretation. As far as symmetric relations such as 'near' are concerned, this language $\mathcal{L}_0$ has adequate expression.

2.2.2 Limiting factors

What happens if the relationship is instead not symmetric? Say that c can be glossed as 'on top of' instead of 'near'. To be able to represent the situation adequately, we

must be able to identify which of a or b is on top. The immediate result is that this is not possible in $\mathcal{L}_0$ with the three symbols a, b, and c. Since the expressions (2.5) are, by assumption, equivalent, as are any strings containing these three symbols, it is not possible to have two separate representations for 'a on top of b' and 'b on top of a'.

One could of course resolve this difficulty by imposing an ordering constraint on the symbols, using, for instance, the first object as the one on top, and the other underneath. In declaring that in $\mathcal{L}_0$, interpretation is not dependent on configuration, we eliminate this possible solution.

Neither can we use a prepositional scheme, nor a post-position 'noun marker' scheme like that of Japanese. This type of representation would involve a separate symbol, say g, which would, in conjuction with one of the others, mark it as taking one of the two available roles, on top or underneath. However, in order to know which object term g is associated with, the two would have to appear immediately following one another, or in some other specific relative configuration, a mechanism banned by our non-configurational assumptions.

Explicitly, we could have '$\_\ g$' be the block on top[3], and so '$b\ a\ g\ c$' would be a representation of a on top of b. But, since configuration is not a factor in interpretation, the string '$a\ b\ g\ c$' must also represent the same relation, even though the pairing '$b\ g$' might be intended to represent b on top.

Having ruled out relative ordering, and the introduction of marker symbols in fixed positions, the only remaining solution is to use different symbols for the components of the relation. This would differentiate their roles within the sentence and ensure that those roles correspond to the roles one wishes to represent. One method of accomplishing this is to multiply the relationship term, using a different symbol for the relationship to correspond to each pair of possible objects. This would vastly

[3]Throughout, I will use the underscore '_' to represent a choice of any symbol.

inflate the number of terms in $\mathcal{L}_0$, and make the user's task of storing[4] them all grossly implausible. In fact, using this method, one doesn't need the object terms at all, as the entire relationship among the three terms is contained within the relation terms.

2.2.3 The complexity of syntax

If we compare the complexity of the language in terms of the number of symbols needed for a given level of expression, we can compare these representations in terms of their economy of storage. If, in the first case, we wish to represent a set N of objects and a set R of binary relations within the language $\mathcal{L}_0$, we need only

$$|\mathcal{L}_0| = |N| + |R| \tag{2.6}$$

total symbols. The language user in this case needs to store (remember) only one term for each object and each relation that could be mentioned. This base estimate is really only a reference point, since the best one can do with this language is talk about unordered, symmetric relations.

In the proposed solution in which each relation is given a separate term corresponding to each pair of objects, we need $|N|^2$ terms for each original relationship $r \in R$, requiring the total

$$|\mathcal{L}_1| = |N| + |R||N|^2 \tag{2.7}$$

This size is cubic in comparison to the $\mathcal{L}_0$ base estimate, and would be completely unrealistic for any practical size set of relations and objects. And still, the expressiveness is limited: only individual (non-compound) objects and binary ordered relations may be represented.

[4] If this is taken as a processing model of natural language, this 'storage task' coresponds to remembering all the symbols of the language, in the sense of being able to recognize and generate them.

Differentiating roles

One of the more difficult problems of using a linear language is the represention of ordered, complex, nonlinear relationships. We could, instead of varying the symbol used for the relation, use alternate forms for each of the objects. In the asymmetric relation 'on top of', we could, perhaps, posit another word, a', which means "a when a is on top of something." We now need a second word for every possible object that could be on top of another. And, if we want to generalize this solution to all other asymmetric binary relations, we need to have an extra form of each object for each relation.

Using this alternate-object approach, we require a set of new object terms N^+ whose size is the size of $N \times R$ which gives us $|N^+| = |N||R|$. A side effect of encoding the relation within the altered form of the object is that the explicit relation term becomes redundant in the expression. We have effectively Skolemized the binary relation, encapsulating the relation and one term into a new relational term which only requires one argument. If a' means 'a when a is in relation c with X', then the expression $a\ X$ becomes equivalent to $a\ c\ X$, and so we can eliminate $|R|$ redundant terms from the language. The functional denotations of these symbols can be written as the following lambda expressions:

$$\begin{aligned} [\![a]\!] &\equiv \mathrm{a} \\ [\![c]\!] &\equiv \lambda x \lambda y.(c(x,y) \vee c(y,x)) \\ [\![a']\!] &\equiv \lambda x.c(a,x) \end{aligned} \tag{2.8}$$

Still, however, we need the original, unmarked forms of the objects to use in the other role of the relation, giving the size complexity of this variant of the language

$$|\mathcal{L}_2| = |N| + |N||R| \tag{2.9}$$

However, this estimate ignores the fact that the unmarked form needs to be associated with some role in the relation. It will take extra processing on the part of the interpreter to realize that this unmarked term should be used to fill in the role remaining after all others have been filled. We could instead mark the role using another set of $|N||R|$ terms, one for each object in the other role of each relation. In this more explicit language, we have even more symbols:

$$|\mathcal{L}_2| = 2|N||R| \tag{2.10}$$

If we attempt to extend beyond binary relations, to include more than two arguments, we must expand our vocabulary still further. For each argument we wish to add to the relations, we need another $|N||R|$ terms to express it with unique symbols. For k-place relations, then, we need a larger number of symbols:

$$|\mathcal{L}_{2k}| = k|N||R| \tag{2.11}$$

In order to express non-symmetric relations, the language user here needs to store a vastly greater number of terms for each object, on the order of the square of the base estimate.

Generalizing marked objects

Still considering only binary relations for the moment, we can generalize this alternate-object approach to an even more economic scheme, that requires only twice the original number of terms. By substituting a small amount of processing, we can reduce the size complexity of the language by a factor of $O(|N|)$. If we assign an arbitrary conceptual ordering of the roles for all the binary relations, then we need only have one special form for each object for each role, but not for each relation. The object given in special form can now be taken as the first of the ordered roles in the relation, while the normal form obect fills the remaining role. This model of role assignment requires added processing on the part of the interpretter of this language, since each object term must now be assigned to the correct relational role, whereas the symmetric relations needed no assignment, and the Skolemized language encapsulated the

assignment so that no further processing was needed. This scheme can be expanded to k-place relations by having k families of special forms corresponding to the k posible relational positions. So with k terms for each object, we have

$$|\mathcal{L}_3| = k|N| + |R| \tag{2.12}$$

which is merely a linear function of our base estimate, and perhaps not too great a tax on the mental capacity of the user.

2.2.4 Generating case markers

This role-differentiating method should appear familiar. It is identical in nature to the concept of case (cf. [Fillmore, 1968]), in which the separate roles of a relation have associated with them high-level abstract categories. While Fillmore has given his categories meaningful names, such as AGENT or GOAL, here we have simply made an arbitrary association with each possible argument of the relation. This could be as simple as an arbitrary ordering: arg_1, arg_2, arg_3, etc. Each argument is then represented by a symbol of a special form, where each possible referent has k special forms corresponding to the k possible case roles.

Generative morphology

One could generalize this mechanism still further, using generative mechanisms to construct the alternate forms of the object terms. In this case, one could have a base form for each object, and a number of productive mechanisms which expand each into a family of individual role-marked terms. This is similar to morphological case-marking systems, in which each single objective term is generatively expanded into a family using morphological differnentiation, giving separate terms for the object in a nominative role, an accussative role, genetive, etc.

This is another example of using added processing in order to reduce the storage capacity needed for use of the language. In this case, a number of generative mechanisms must be used in order to eliminate the enormous number of unique base terms needed for the general case-differentiated scheme.

However, one must be careful, that these generative mechanisms don't defeat the spirit of the non-configurational assumption. Affix morphology is a common system appearing in real languages, in which a small stable set of affixes are used in conjunction with a base form to generate a case-marked variety. Analyzing the language at this level of base-forms and affixes, we see that each affix must be adjacent to the base form in order to be associated with it. If we were to uphold the non-configurational assumption at this level, such a mechanism would be in viiolation, since re-ordering the base and affix, or some combination of bases and affixes, would destroy any identifiable relation between them. This kind of morphological solution is merely a shift of focus, from what I was calling 'terms' above, to 'base-forms' and 'affixes'. It still relies on juxtaposition to specify the relationship. Eliminating this relative ordering would defeat the functionality of these generative mechanisms — an accussative suffix won't do much good if there is no mechanism for identifying its base term, regardless of intended meaning.

In order to avoid configurational rules then, the differences in role terms must be identified by means other than simple affixation. There do exist real morphological mechanisms, such as vowel shifting, that avoid this difficulty, while still retaining the space advantages of a generative role-marking procedure.

2.2.5 Sentential problems

Even if we assume a complex, case marking morphology, many problems remain in this asyntactic language $\mathcal{L}$. A particularly problematic question among them is the

problem of delimiting relations. So far, I have discussed only cases with one relationship along with the objects participating in it. What happens when a language user wants to mention more than one relationship? Some mechanism is needed so that the various relationships are kept separate in interpretation, and that the components and specifiers of one relationship do not get mixed up in those of the other. This division, although somewhat ill-defined, is normally attributed to the clause or sentence level of syntactic analysis.

The most obvious way to define a sentence in $\mathcal{L}$ would be to have objects participate in the closest relation for which the role is not filled. While this mechanism does not enforce any strict arrangement on the objects, it is still a configurational constraint, since it will not allow arbitrary ordering of the terms involved. If each object term is associated with the relation closest to it in the sequence of symbols, then if two relations are intermixed heavily enough, the object will participate in the other relation, and not in the one for which it was intended.

At first thought, it seems that the time domain could be brought into play here to resolve this difficulty. If a pause, or some other change in timing is used to indicate the separation between one complete relation and the next, then could one claim that the language is non-configurational? One could still arbitrarily arrange the symbols of one sentence, as long as they remained between delimiters. Ultimately, because the pause separates those symbols that come before it from those that come after it in sequence, it is still a constraint on the configuration of those symbols. It is not a constraint between the object terms and the relations, however, but is a constraint between the pause and all other symbols, in disallowing relations to be built across pauses.

Our language will not be able to support multiple relations under these assumption using either the unmarked or the case-marked object terms. Since neither of these forms carry information about which relation they are to take part in, the possibility of confusion always exists when more than one relation is used. Because the Skolemized versions of the object terms carry the identity of their relation with them, multiple relations using this mechanism will not become confused, at least up two a point. When reference needs to be made to two relations *of the same type*, however, it will no longer be able to differentiate between the objects of those two. And still, this solution carries the enormous size burden of requiring a separate form of each object term for each relation that is expressed within the language (not simply within a single statement, as with the case-marked version).

2.3 Benefits of configuration

This exploration of the limits of a non-configuration language may seem an absurd reduction of real languages, all of which carry configurational constraints. It serves, though, to highlight the direct connection between configuration, representational capacity, and language size. Having a language with absolutely no configurational constraints obviously places an enormous burden on the language users in terms of the size complexity of the language. Even as simple a constraint as delimiting relations with a pause, punctuation, or similar symbol allows us to use a symbol set that is exponentially smaller than the non-configuration approach of the same representational capability ($O(k|N|)$ for the case-marked symbol set versus $O(|N||R|)$ for the skolemized).

The configurational constraints also reduce the amount of storage and coordination necessary to process the language. Let us use just the storage required to process one relation as a measure of processing complexity. Looking at the relations made up of triples of symbols (a_i, b_i, c_i) in example 2.13, we see that the case-marked language, with relational delimiter, requires only as much storage as there may be symbols in a single relation. When a delimeter is reached, all the symbols since the last delimiter

are known to comprise an entire relation, and no further input storage is needed.

$$a_1\, b_1\, c_1 \mid a_2\, b_2\, c_2 \mid a_3 \ldots \tag{2.13}$$

$$a_1\, a_2\, a_3\, \ldots\, b_1, b_2\, \ldots\, c_3\, c_2\, c_1 \tag{2.14}$$

In the skolemized case, however, an arbitrary number of symbols and relations may intervene between the symbols of any given relation (a_1, b_1, c_1 in this case). Human language users obviously do not have infinite memory for symbols, and therefore configuration *must* be part of the interpretation process. Simple constraints such as the relation delimiters have an enormous effect on the efficiency and complexity of language processing.

More specific constraints

Other constraints on configuration also have an effect on the size and processing complexity of language. Consider an operation such as logical 'and' ($\wedge$). This operation is normally used to construct a complex object out of two base object terms. Consider the two versions of the same relation in example (2.15), where the intended interpretation is $r(\,(a' \wedge b'),\, c')$; that is, a and b together participate in relation r with c. What mechanism will allow us to distinguish a' and b' as the components of the 'and' operator?

$$\begin{aligned} &a'\, b' \,\wedge\, r\, c \\ &a'\, c \,\wedge\, r\, b' \end{aligned} \tag{2.15}$$

We could again postulate a multiple term solution, in which separate symbols are used to represent a when it is a member of the 'and' operation and when it is not. This would require a separate 'and' form for each of the case-marked forms that already exist. If other sub-relational operations like 'and' are used, a separate form must also be postulated for each, in a parallel fashion

Again, configurational mechanisms require less language size and processing storage. We could use a subsidiary delimiters, such as left and right brackets [and], which would bound the components of the sub-relation and separate them from the remainder of the relation. This eliminates the need for a multitude of new symbols, and requires only linear processing storage. Or of course, we could use the familiar solution, in which the operator is constrained to relate only those two objects appearing immediately preceding and following it. This solution requires no new symbols at all, and does not increase the required processing space.

2.4 Representing structure

These arguments are not intended to claim that syntax is the result of only semantic considerations. Rather, the claim is that the use of ordered syntactic constraints on representation have obvious advantages of storage and processing efficiency, for the goals of representation and interpretation. These constraints allow the semantic relations between symbols to be displayed in their configuration, rather than requiring that the individual relationships between symbols be known by the language user. A rich variety of possible relationships is permitted using far fewer distinct symbols than any of the non-configurational alternatives.

Transparent semantics

As a consequence of using these configurational constraints, some of the semantic properties of the relationships become transparent. If a particular semantic relation is always expressed using the same configuration of its object and relation terms, one can observe, *by studying the configuration alone*, which symbols are allowed in the relation along with their respective roles. This is the basic assumption underlying any attempt at distributional linguistic analysis, whether it is based on numeric probability or

symbolic grammar induction techniques.

Harris suggests [?] "morphemes can be grouped into classes in such a way that members of a class have rather similar sets of co-occurrents, and each class in turn occurs with specific other classes to make a sentence structure." It is not far from this to assume that if a word is observed to be used within a particular set of syntactic configurations, then it shares semantic features with other words that appear within the same set. Hindle and Rooth [?] show this in a model of prepostional phrase attachment, in which co-occurrence statistics are used to decide whether a prepositional phrase modifies a preceding noun or verb. The decision model uses appearance data "as evidence of an underlying relationship, but makes no attempt to deteremine what sort of relationship is involved."

This result is somewhat surprising, because it implies that we can observe these properties *without requiring a semantic model.* Transitively, if we assume (as we have been) that the symbols make reference to internal concepts, this regularity of configurational appearance allows us to observe which concepts are capable of being related, and thus to observe semantic properties of the symbol referents. It does not require a specific theory of semantic properties and their interactions to note which terms are used in conjuction, and to discover which configurations never appear. It also implies that semantic properties can be attributed without having a detailed model of syntax, and without requiring a parsing theory which decodes sequences into semantic relations.

Having a semantic theory certainly does not impede the method. Pustejovsky et al. [?], for example, use data of a term's appearance within a syntactic relationship to support an assignment to it semantic properties which license that relationship under the theory. They also propose that appearances of a term within a structured collection of syntactic configurations is sufficient evidence to posit for the word a detailed semantic structure licensing those configurations.

Syntactic–semantic interactions

That the syntactic behavior of words is determined by their meaning is of course not a startling revelation.

5

[Summary?]

What we see from studies such as these is that word order and represented meaning are highly interdependent.

[much more summary needed here!!]

2.5 culls

This view of the nature of grammar has gone in and out of style over the years. Central to Bloomfield's notion of linguistic structure [Bloomfield, 1933] was the notion of the *sememe*, which went hand-in-hand with the *morpheme* as the minimal representation of a meaning connected with form. By the time distributional syntax was well developed, the widely held position was that formal grammar could be independent of semantic considerations (see [Wells, 1957]). Chomsky, in [Chomsky, 1957], takes the extreme position that "any search for a semantically based definition of "gramaticalness" will be futile," but much later [Chomsky, 1965] seems to reverse this viewpoint by postulating 'deep structure' (constituting a 'semantic interpretation') as a fundamental component of syntactic description.

No doubt some of the reluctance to include semantic considerations in a formal description of language is due to the difficulty of observing meaning itself. Words can be seen, heard, counted and measured, and are thus nicely amenable to formal description. Meaning is much more difficult to observe, much less represent.

Of course, this is not to say that a distinction between syntax and semantics is not

[5] cite Zernik and Mats Rooth wkshp paper on clustering

warranted, nor that it is not useful. It can be greatly useful. However, it should always carry a reminder that any distributional regularity, or syntactic systematicity, will go hand-in-hand with a semantic regularity. Any pattern or grammar rule describing a class of syntactic configurations can do so only by reference to classes of terms and sets of strings in the language. It is this regularity and predictability that a grammar is designed to capture, both in terms of the arrangement of words in the sequence, and in their roles relative to one another.

If those rules of grammar are written with respect only to base strings of the language, for instance if the description is nothing more complex than an enormous listing of KWIC[6] concordances, than that grammar has very limited descriptive power. The same can be said of distributional descriptions with no structural models, such as the recent trend toward hidden Markov models. Without being able to abstract over classes of words, or model the interaction at a level higher than individual occurrences, they really tell us nothing about why the language is structured the way that it is, or how it operates. The analytic power of a grammar comes from abstractions made over classes of words displaying similar behavior throughout the language. As soon as one admits these types of classes,

[why am I harping an syntax v. semantics?]

From this point of view, the syntax v. semantics debate can be summarized in the following way: do these constraints governing which words may appear together, and in what configurations, stem from causes separate and different from any constraints which might govern what the interpretation of that configuration of words might be? Under the present assumptions, this becomes a chicken-and-egg debate: a given syntactic configuration of a sequence is allowed if it has meaning, and meaning is derived through the configuration of a sequence. The two are differenct, and can be thought of separately, but neither can exist without the other.

[6] Key Word In Context.

The differences seen in these examples is not because of a difference in *kind* between syntactic and semantic constraints. The difference lies in the *level of generality* of the properties of the language that are being violated in each case.

[this is really the point, isn't it?]

One possible approach to answering this question is the consideration of the links between the two. If words are constrained as to their meaning, and combinations of words are constrained as to their configuration, then it follows that the representation of compound meanings is also constrained as to configuration. That is, constraints on word configuration will constrain the manner in which complex meaning may be assembled out of words, regardless of any inherent limitation in the manner in which meanings may combine.

It is not so much that we cannot conceive, discretely, of an act of making something stronger, nor even that this is something desirable in making a soup, say, but that this act, as referred to by the *word*, 'strengthen', comes with mechanical limitations on the methods by which it may combine coherently with other concepts.

2.6 Associative features

The previous section showed that word use is not arbitrary, but rather that it is highly systematic within the language. The variety of uses of words are interrelated, and, after observation, somewhat predictable. The fact that we as language users are able to make judgements about word choice and acceptibility, especially about novel constructions and nonsense examples such as the *gardle* sentence (2.3), shows that these systematic relations have a real effect on the interpretation of language.

One of the primary goals of any grammatical description is to capture this sytematicity in word use. These generalizations are typically encoded in terms of features

such as phonology, morphology, part of speech, case, number, gender, attachment preferences, as well as through semantic attributes such as case roles, argument structure, and selectional restrictions. One consistent theme in all of these linguistic descriptions is their reliance on *interpretive* features. With the possible exception of phonology, none of these properties is at all apparent in a word itself, but rather all are secondarily derived through our existing knowledge of the use of the word. This knowledge derives from our competence as language users — that in our ability to use these words in a meaningful way, we have knowlege of some of the regularities of their use. We have been able to categorize these properties into the various types, and to differentiate words based on their values within those types. This ability, to interpret our own linguistic ability in order to capture the regularities in our own use of language, is what brings power to these grammatical descriptions.

[7]

However, one common difficulty of these systems stems from this same source as this power. Each relies on knowledge derived from our experience as language users, operating in an extra-linguistic environment. Because we use language in a communicative and referential capacity, our semantic knowledge of the connections between words and their communicative function comes to play in any attempt to characterize the use of those words through introspection. The semantic relations are always the final arbiter of acceptibility, in that if an utterance, a string of words, is not or cannot be understood to hold meaning, then it is not, and cannot be, accepted as a viable statement in the language.

The difficulty this holds for descriptive linguistic systems is simply that the representation of this meaning, these semantic relations, is extremely difficult, as it does not

[7] need to reorganize some of this section¿ The structure exists on a large scale, why throw it away with 3-gram HMMs? Need a model that explicitly represents these considerations.

Need a better contrast with the MM crowd. w.r.t. the n essential components of a linguistic description, how do HMMs fall short? And how can we better represent these in an informative fashion?

exist in a readily tangible form. These interpretive data are not directly observable, but exist only within our minds.[8]

2.6.1 Language Data

Much of the data one finds in linguistics literature today is subjective and intuitionistic in nature. Examples sentences are given, like those of the previous section, along with the author's (and rarely others') subjective opinions as to their aceptability on various grounds. These examples, sometimes terse and violent[9], and perhaps unlikely to be heard in actual dialogue, are usually constructed in order to draw attention to very specific structural distinctions between strings of words that are members of the language and strings that are not. By constructing groups of examples which exhibit most all of the same grammatical properties, the authors hope to identify more exactly just those factors which affects the examples' acceptability. One sentence may be deemed aceptable, but the addition of a small modification renders it unacceptable. Such modifications stradle the border between the space of acceptable usage and that of unacceptable use. In studying this border, the hope is to clarify the structural and grammatical priciples that define the member strings of the language.

The focus on data generated in this manner suffers from two problems. First, it often reflects only the language as used by the author, whose idiolect may not correspond with the language used by the larger community. Especially when an author focusses on a more subtle point of grammar, there can be considerable debate over the generality of the acceptibility judgements, which brings into question the foundations on which the author's reasoning rests. The question arises, 'is this a general phenomenon, or is it an idiosyncracy of the author's available knowledge of the few words used in the examples?' People are much more adept at recognition

[8] And by extension, out heads. There is an increasing body of work studying brain activity as it relates to language skills. [?]

[9] isn't there a Pullam article (in the eskimo book) on the violent language in example data?

than at recall, and the examples available to the author through introspective recall may not cover the knowledge available for recognizing and judging numerous other examples. One might be better studying a set of relevant utterances as used by others, which is likely larger, more diverse, and a more complete example of usage than any set derived through introspection and recollection.

The second and more serious disadvantage that such short lists of examples carry as the foundations of linguistic argument is that they rely on an appeal to the reader's sense that the examples represent a related set of phenomena, which vary importantly only in those aspects on which the author is focussing. This is an appeal to the reader's introspective sense of the structure of the language, an argument that the categories and generalizations as seen by the author are really the same as those the reader has. There is no impartial presentation of the larger set of language data, against which to verify the analysis.

The short lists of examples used for analysis can hardly be considered truly representative of whatever underlying phenomenon is being described. The favor that lexicalist theories and representations have currently obtained is evidence to the fact that all words have idiosyncratic properties, and that none are completely generalizable. A paradigm represented by a list of five or ten examples (if that) in a paper may evoke an agreement in a reader's understanding of the language, but in truth, it may only represent those examples given, and not indicate a general pattern at all, but a phenomenon specific to just those few words. This is not to say that such studies are not valuable, but that without data supporting the generality of the phenomenon, a larger argument is needed in order for them to speak beyond the particular examples given.

[However, with that in mind, let us move on to a few examples.

2.6.2 Some examples

Start discusson with ONE WORD in typical context Show it in other contexts analyze the failures in terms of meaning type (semantic category) syntactic type i.e. - various interpretations of the failure. Reason 1.that this particular wor can't show up arbitrarily in text, which can be used to imply 2. that there are constraints on how it may be used (correctly..)

show enough examples of good use to begin to characterize the usage show some examples of the word used with a fairly different meaning, to catch people off guard show that the porblem is even more complex

Consider the word 'gold' in the following construction (1):

(1) The bar is made of gold.

This is a very simple construction, syntactically, placing limited constraints on the word appearing in location of 'gold'. There are a great number of other words we can use in exactly the same construction. 'Wood', for instance, is perfectly acceptable, as are 'maple', 'poly-butylene', 'steel', and the generic 'stuff'. There are some words, which are not quite acceptable in this construction: 'light' cannot be used here, nor can 'humor' or 'linguistics'. Other words are questionable in such a context. 'Water', for instance, is not a substance one would think of as composing a bar, but if it were frozen, or formed the bulk of this particular bar, one might be able to make sense of this statement. There are many words which fall into this intermediate category, words which are not the first things one might think of as filling the role, but words for which a little thought or creativity allow them in the position.

Let us look at an even less constrianed use of the word 'gold':

(2) That fellow has a lot of gold.

Every word which can be used in (1) can also be used here in (2). Further, every word which is questionable in (1) can be used with no hesitation in (2). Even many of the words which are not acceptable in (1) can be used in (2): 'light', 'humor', and

'chutzpah' are good examples.

What this analysis is missing thus far, however, is the simple observation that *most* words are unacceptable and cannot be used in either construction. 'Instance', for instance, is unacceptable in both. So are 'serious', 'is', 'eating', and 'acceptable'. As general and simple as these two constructions are, the great majority of words fall into this last category, being unacceptable in either context.

Why should this be so, that only certain words can be used in this context? And how do we know, for any given word, whether it could be used in either of these contexts, when surely we haven't had a chance to think about this particular example before reading it here?

[much more here, about what the problem is, and what the nature of lexical knowledge is, before moving on. Remeber to point out the constrast between clearly unaceptable words, and questionable 'close' words like 'water'. (are bars made of water? semantic, not syntactic ground. But is it just a matter of degree, and not kind?)

2.6.3 Parts of speech

Even grammatical features as basic as part of speech, although sometimes taken for granted in linguistic analysis, are not intrinsic to the language, but stem from a classification of words as they are used. Recent work in part of speech tagging has verified that tags are a language property that can be learned by machine. Church [?], and more recently Weischedel et al [?] have proven tags to be decidable by Markov models to near-human accuracy. Brill [?] has constructed a rule-driven procedure based on word identity and context that decides traditional POS tags. These developments suggest that whatever features we as language users exploit to identify part of speech, or at least some functional equivalent to them, are derivable from the word usage data, without resort to semantic information.

However, it leaves open the question of whether the part of speech tags are really the set of abstract features which one wants to use in language processing. Certainly, there is a large gap between the generality of POS tags, and the sets of descriptive features used in, for instance, case grammars, or lexicalist representations such as the Generative Lexicon [Pustejovsky, 1995]. If the POS tags are derivable from, and therefore verified by, the available data, why not these more detailed and meaningful representations?

The parts of speech are themselves chosen on an 'as needed' basis. Derived from historical studies of now-dead languages, parts of speech are used as encapsulations of grammatical properties, constraining the possible grammatical constructons in which their members may participate. In most theories of English grammar, a noun, for example, is allowed to be the subject of a sentence where the remainder is a verb phrase. A noun may also combine with a determiner and an adjective phrase to form a noun phrase as the subject of that same sentence.

The process is not always that simple, however. Some nouns are found as the subject of certain classes of verbs, whle others are not. "Verve", for example, or "restitution", are perfectly acceptable nouns which do not exhibit normal behavior – for instance, they will not be found as subjects of a large majority of English verbs. The traditonal classification made by the parts of speech, or even by complex schemes such as that used in the Penn TreeBank or Lancaster studies, does not make fine enough distinctions to resolve situations such as these that are prevalent in the language structure.

Part of speech is often assigned in ambiguous cases in order to encode or justify the analysis that has been made.

[10] The question arises, then, of how empirical a study along these lines can be.

[10] this is a completely ad hoc argument. Or is it simply rhetorical support for producing this kind of study?

How much evidence exists in a sentence itself to support a given grammatical analysis, andd how much external knowledge does the linguist add that is not encoded in the theoretical rules and representation?

The existence of these classes is not in dispute, but their number and degree of separation has not been well measured. [11] These classes have traditionaly been assigned based on observation and a pre-existing (semantic) understanding of word function as it affects and defines the meaning of words in context.

Indeed, a cognitive constraint argument based on computational limits leads one to believe that there will be classes of functional types that the functional components of language will fall into. By looking at the distributional properties of language, we will find some of those classes.

By attempting to make such classification without this knowledge we can verify its independent existence, and also gain insights into how such a classification of arbitrary symbols can be acquired by an uninformed learner.

2.7 Observing language structure

If the regularity in the syntactic and semantic structure of language is truly present, we should be able to find evidence of it in the observable artifacts of language — in the spoken sounds and written words we use to communicate. One of the main difficulties in observing these regularities, however, is the nature of the language itself. Linguists, as natural languge users, are able to study both sides of the problem, the symbols along with their meaning. The theories and data sets, however, are not a natural phenomenon to which humans have evolved a natural facility, and are thus understanding the process is much more difficult than experiencing it. Theories have grown to such a size and complexity that the implications of new data or even

[11] Are there *any*?

small changes to the framework can be far from obvious. The growth of corpus-based computational linguistics is a direct result of this fact, in part providing methods of automating the task of checking a theory or representation for accuracy and coverage.

The machines, however, are at this point only able to view one side of the problem: the symbols. The representation of language semantics lags far behind the direct manipulation of symbols and their occurrence in sequence. New methods and new characterizations that are capable of describing language without incorporating meaning, but which still leave the researcher avenues for meaningful interpretation, will have to emerge to make this new type of automated language study complete.

Traditionally, language has been studied from an introspective viewpoint, one in which the meaning behind the words is assumed to be understood. Some linguists, certainly many syntacticians, would balk at this assertion, but let me present just a few examples to clarify how meaning is intertwined with the study of language. Classification of verbs and prepositions based on the 'event structure' of the underlying meaning is a recent case in point; often, the phenomena are given syntactic interpreations.[?] However, without a clear knowledge of the semantic implications of the phrase under study, one cannot even begin this task. As a second point, consider that the acceptability judgements ('starred-ness') of sentences is often judged with respect to the intended meaning of a phrase. Taken one way, the utterance may be perfectly acceptable, but it may be judged syntactically ill-formed with regard to some different meaning.

Even the study of syntax has not completely divorced itself from semantics.

[Remove this talk about Chomsky. Read up and present an intelligent discussion of some of the methodological hypocracy of syntax]

The Chomskian government and binding school (GB), who are some of the major advocates of semantic avoidance, still incorporate a great many semantic notions in their theory and analysis. The notion of 'deep structure', for example, is one which

implies an underlying representation for the surface phenomenology of language symbols. The supposition of traces, and of constituents, such as INFL, which may be absent from the surface structure (the lexical symbols), also implies a

The structuralist school shied away from this to some degree. Their agenda consisted, in part, of determining *substitution classes*, terms or phrases which could appear in identical contexts and still be acceptable to a native speaker. Of course, the notion of acceptable still falls back on whether the resulting utterance has some possible interpretation. Phrases are tested for substitutibility not in the abstract, but with repect to the thoughts they convey; No linguist would consider asking if one could replace

[Conclusion: we would like to characterize the **formal** structure of langauge, and so we should look at the observable evidence and postulate from there.]

2.7.1 Visible associations

If all we know about words is how they appear in sequence, and we're assuming that configuration contributes ultimately to 'meaning', how can we describe/extract those features of the configuration that play a part?

1) distance limited. One can only keep track of so much stuff — at some large distance, the average interactions between words must die off.

This implies that words used in collocation will constrain the meaning represented more than words used at a great distance (on average).

This does not rule out occasional long distance interaction, but relegates it to special circumstances. Long distance interaction will not be the norm, but will exist when very particular meaning constraints must be satisfied.

2) The number of words used by a typical langauge user is large (50-100k–ish, get some real figures). They can't all work independently in terms of syntax – too many

combinations to keep track of, as well as the problem of mutual interactions: if A affects B, does B affect A, and how do you know what the effect should be? There must be classes of words that behave at least similarly, if not identically, in terms of syntax. (cite all the structuralists – Bloomfield, Harris (then& now), Langacre)

[this is probably the most important bit in thsi section!]

Conclusion:

1) for given word, the words appearing locally with it will appear only in certain configurations, based on their classes. Those configurations will be correlated with the meaning of the term in question, that is, particular configurations will group more with particular meanings than with others.

2) conversely, words of the same class will have similar configurations that appear with them.

2.7.2 The available data

Although the conclusions may, in the end, appear obvious, I believe a short discussion of the nature of this available data is warranted.

The directly observable data in language studies is generally taken to be comprised of the speech and writing used in communication.

Only one small component of the language system is easily available for direct study — the physical symbolic component. /the written artifact. [12] Spoken language can be recorded, written language is by its nature artifactual; thus they are ideal subjects of direct study. The other aspects of language: internal thought processes and memories, along with the external world being referenced, are much more difficult to capture [13].

[12] Signs, signifiers, significant, etc.

[13] Should make reference to both of: discourse studies takng environment and context into account, and direct brain response studies (Squids, Hall-effect sensors, etc.).

A great deal of such data is available, giving numerous examples of language as it is actually used. A mind-bogglingly unanalyzable amount of language data exists, written in books and papers, recorded on audio and video tape, and stored electronically using computers. The recent growth in the field of corpus linguistics reflects the trend toward studying this immense supply of objective, positive data.

A written language is usually quite different from its spoken form. They use completely different symbol systems — acoustic signals in spoken systems, and graphical signals in written language. The symbol systems rarely coincide exactly; one morpheme may have more than one written counterpart (homographs), or differentiable sounds may have the same written representation (homonyns). Some written languages have the advantage of symbols and structure not representable in their spoken counterparts. However different the two may be, written language provides an ideal source of data from which to study language structure, one in which many features of the language structure have been made explicit (see [?] for a lengthy comparison of the organization of text and speech).

not available in spoken

- **Distinct word boundaries** — word boundaries are clearly marked by whitespace and punctuation in written text. Detecting word boundaries in a speech signal is still the subject of significant study.
- **Consistent word representation** — The same word is always writen in the same way (barring spelling errors or varying spelling conventions), whereas spoken words may differ greatly from instance to instance, and even more greatly from speaker to speaker.
- **Punctuation** — with the lack of important stress, tone, and timing information, puctuation serves to provide explicit clues as to the connectivity in grammatical constructs.

- **Discrete symbols** —
- **Non-linear spatial organization** — Because of it's two-dimensional layout on the page, written language has an extra dimension of proximity to expoit, against the strictly linear temporal ordering of speech. Titles, headings, paragraph layout, and tables and charts are examples of the exploitation of this added dimension.

There are, however, many features of spoken language which do not have counterparts in the written parallel. These features can provide a wealth of information about the intended meaning of the speaker, as well as carry other supplemental information not present in the simple string of words.

- **stress** —
- **tone** —
- **timing** —

2.8 Summary

need a conclusion.

Reasons

Why is this sort of study (explicitly studying word-context relationships) useful?

1) Learning – it's more direct to study and learn the direct associations of a word in use than it is to try to build grammars of the entire language. The detaile model of the data is more irect and easir to study and understand.

2) It avoids having to know about the trees. Much lnguistic study is meaning-oriented (even syntactic study), and requires that one incorporate knowledge about

phrasal constituents and syntactic level. Dirctly studying the associaiton of a word to its context avoids having to know about, or postulate theories about the form of, this higher level language knowledge.

3) potentially more useful in complex, reeal world (i.e. outside the bounds of the data that was used to form the model) situations

Chapter 3

The Associative Model

The last chapter introduced the dual focus of an associative model of word use. The first goal was to investigate the constructions in which a word may appear. This goal can be restated as an attempt to characterize the constraints between a given word and all its possible contexts. The second aim was the converse of the first: if the context and the word are mutually constrained, how does the choice of context, or construction, limit the choice of the words used within it?

These are questions of both representation and abstraction. In order to characterize a set of constructions, one first must decide what actually constitutes a construction, and how much of that construction is reasonable to include for study. One must also decide how to represent it so that it is comparable to others.

The abstraction comes in characterizing the set of words or constructions versus the individuals. Listing each and every possible context in which a word might be used is certainly a complete method of characterizing the constraints, but it is hardly useful for gaining a better understanding of the system of linguistic constraints, nor is it useful as the underlying structure of a processing system. Some summarization and abstraction is surely needed, either in the form of statistical distributions, or terms of categorial assignments.

This chapter sets the framework for building and using such an abstract categorial representation. The model I propose is based on categories both of words and of constructions, where each word has partial membership in a number of lexical categories, based on the constructions it appears with, and where constructions belong to categories based on the words which comprise them.

The first section deals with the nature of individual constructions, and how they may be chosen, represented and used. The second section covers how comparisons may be made within the representation, to describe the relations between parts of the language system. The later sections describe how generalizations and abstractions can be made consistently across the underlying data, in such a way as to preserve the relations between the individual words and constructions. The following chapters cover the specific techniques used, and describe some of the derived results. [1]

3.1 Describing usage

3.1.1 Diverse usage

The variety of constructions in which a single word can appear is not apparent *a priori*. In fact, the diversity of roles available in the language as a whole is also not well-defined. One might use well-known categories such as the various parts of speech to separate the roles. This can prove difficult, however, do to conflicting intuitions about what the use and interactions of these categories. Because we have chosen to use only the occurrence data of the language, and are trying for the moment to ignore the fact that indeed, as speakers of the language, we can identify and discriminate

[1] Questions to be addressed:
1) How does the model represent the important properties of inter-word constraints, categories, etc., which (we hope) were outlined in ch2?
2) how does this model reflect on the weaknesses of HMMs?
3) Is this as useful or as rich as conventional syntactic models?
4) is it complementary to normal syntactic description?

between, say, adverb and preposition, we have no way to label the variety of usage a word might exhibit in terms of such a set of external categories.

We are relying here on the internal evidence given by the language data to suggest to us some set of categories, which, in the end, we can compare to the familiar set that has proven to be helpful in traditional linguistic analysis.

Word centered models

A number of classification schemes exist which seek to perform a similar assignment of words to funtional classes, with good success. [?] and [Brown *et al.*, 1992a], for instance, present some admirable results. These methods and others, however, group words according to the total of their observed usage, ignoring the fact that a single word may exhibit a variety of different behaviors within the language.

Assuming, within the model, that all used of a word are related may be premature. If we as linguists have already recognized that individual words have many meanings and various modes of use, it seems a step backward to consider each word as a monolithic entity, as in these word-based classifiers. To compare a word's total usage against that of others, without investigating some more basic structure, is to deny well accepted linguistic generalizations.

We would like instead to build a representation of the *ocurrences* of each word, and decide from that how those uses are related, and in what way. In studying the *individual appearances* of each word, and comparing each construction that the word appears in, one can build a composite description, which will identify and discrimate between the various modes of use which the word takes.

3.1.2 Context data

Without a theory of word use in language, the most complete characterization of a single appearance of a word, in terms of the textual data we have restricted ourselves

to, would be a description of the entire context of the text in which it appears. This could be construed as the 'maximal construction' of which the word is a part. If we consider a given text corpus, t, as a sequence of N symbols, $[t_1 - t_N]$, this general characterization would situate the symbol t_i within the context created by the sequences extending to either side of t_i, given as:

$$[t_1 - t_{i-1}] \quad t_i \quad [t_{i+1} - t_N] \tag{3.1}$$

A compact description of a use of a word could simply be its position in such a corpus. This simple representation, though, would not be at all useful for explaining the variety of occurrences, or for reasoning about the use of the word.

Following the structuralist assumptions laid out in the previous chapter, we are asssuming that the form of this sequence is in part constrained by the identity of the item at t_i. That is, the nature of the interrelationship between the symbol at t_i and the elements at other positions in the sequence limits the possible configurations occurring around t_i.

If we admit the argument that these interactions are primarily local, we can, without loosing much information, restrict ourselves to a smaller window of local interactions, and use only the immediate neighbors of t_i. Instead of the entire corpus sequence, we look only at the immediate neighborhood. Specifically, we consider the sequence of tokens extending n positions before and m after the token under study:

$$[t_{i-n} - t_{i-1}] \quad t_i \quad [t_{i+1} - t_{i+m}] \tag{3.2}$$

This characterization defines a general notion of lexical context, which consists of a specification for the immediate neighbors of a word in use. If we re-index the tokens so that the central position is t_0, we can write this context c as:

$$c = [t_{-n}, \ldots, \; t_{-2}, \; t_{-1}] \quad t_0 \quad [t_1, \; t_2, \ldots, \; t_m] \tag{3.3}$$

This general context specification, with n symbols extending to the left and m to the right, I will call an n, m-*context* of the word w at position t_0.

3.1.3 Context properties

[I'm trying to explicitly represent the constraints a word places on the context, as well as those placed by the context on the word.]

By specifying any particular context c_i, we also specify the set of words which occur as the central element t_0 of that context. Within the corpus, let $\{w_j | t_0 = w_j \text{ in } c_i\}$ be those words w_j which are used in the specific construction $c_i = [t_{-n} - t_m]$.

This set w_j comprises a subset[2] of the words which are allowed to appear in the context c_i, according to its structural constraints. Knowing this set of words for all observed contexts gives us a basis for comparing the constraints operating within each construction. Two or more contexts which appear with identical, or similar sets or words could be inferred to project similar constraints. Decomposition and identification of basic constraints could be obtained then by observing the variability of word use across contexts. If we see a context, or group of context, that is used only with a subset of those words appearing with another group, we can assume that the first group imposes a tighter set of constraints (since fewer constraints allow greater variety of words). By noting the overlap and boundaries of these word sets, we could then construct a fairly detailed delineation of the constraints operating between word and context.

For a more detailed analysis of the strength and boundaries of those constraints, we can also use the frequency with which words appear in a given context. A word appears very frequently should be highly associated with the constraints of that context. Let us call the count of w_j within the context c_i, relative to the number of occurrences of the context with the entire corpus, the *word frequency* of a word w_j in a context c_i. This relative normalization will allow us unbiased comparisons between different

[2] Due to size limitation on the sampling, the set of words observed in this context may be far smaller thatn the set which a speaker might find acceptable.

contexts. We write the *word frequency* of a context with respect to a corpus t as:

$$w_j(c_i) \stackrel{\text{def}}{=} \left. \frac{\|\{c_i | t_0 = w_j\}\|}{\|c_i\|} \right|_t \tag{3.4}$$

For a fixed vocabulary with size V, we can use the word frequency of every word with respect to a single context to represent the constraints associated between this context and the entire vocabulary. We can express this *word frequency distribution* of a single context as the vector:

$$W(c_i) \stackrel{\text{def}}{=} [w_1(c_i), w_2(c_i), \ldots, w_V(c_i)] \tag{3.5}$$

Because we have normalized by the total ocurrences of the context, this frequency distribution sums to 1, and can be treated as a probability density.

$$\begin{aligned} \sum_j w_j(c_i) &= \sum_j \frac{\|\{c_i | t_0 = w_j\}\|}{\|c_i\|} \\ &= \frac{1}{\|c_i\|} \sum_j \|\{c_i | t_0 = w_j\}\| \\ &= 1 \end{aligned} \tag{3.6}$$

This representation of context contraints, as a set of words and relative frequencies, gives us a nice account of the interactions between constructions and words. It is, however, tightly constrained to the available data sample over which the frequency counts are taken. (Section ?? explains in detail why this is such a large concern.) In order to make predictions for the behavior of constructions we have not observed, or to draw generalizations about the nature of the constraints operating within the language, we will need analytic methods of comparing separate constructions, and of predicting their similarities and differences. As these constructions are composed of words, a compositional account requires also a model of word use, and the interactions between individual words.

3.1.4 Word Usage

Using this representation of context, to understand the interplay between a certain context and the remainder of terms in the language, we now need a representation of the relation between each word w_j and the contexts with which it might appear. Analogously to the way in which we describe the dependency between the context and the choice of possible words with which it can occur, we can reverse the roles, creating a dual model which descibes context choice as a function of the word.

Specifiying any given word w_j picks out the particular set of contexts in which it has been used (w.r.t a given corpus, of course). We can write that set as $\{c_i | t_0 = \mathrm{w}_j\}$, the set of all c_i where w_j is the central element.

Again, a more detailed account will also require the frequency with which the word appears in each given context c_i. Some constructions may appear much more frequently with one word than others, indicating perhaps that the word has a high affinity for those constructions, and more easily satisfies their constraints.

Let us call the frequency with which a context c_i is used with a given word w_j the *context frequency* of w_j. This can also be normalized with respect to the total number of occurrences of w_j in order to allow unbiased comparison between the relative frequency of contexts with respect to different words. We write this normalized context frequency as:

$$c_i(\mathrm{w}_j) \stackrel{\mathrm{def}}{=} \left. \frac{\|\{c_i | t_0 = \mathrm{w}_j\}\|}{\|\mathrm{w}_j\|} \right|_t \tag{3.7}$$

For the complete set of possible contexts, $c_i \in C_{n,m}$, we can collect the frequency of every context with respect to a single word and use it as a representation of the association between the word and the constructions in which appears. We can express $\mathrm{C}(\mathrm{w}_j)$, the entire *context frequency distribution* of a single word as the vector:

$$\mathrm{C}(\mathrm{w}_j) \stackrel{\mathrm{def}}{=} \left[c_1(\mathrm{w}_j), c_2(\mathrm{w}_j), \ldots, c_{C_{n,m}}(\mathrm{w}_j), \right] \tag{3.8}$$

Because we have normalized by the total ocurrences of the word, this definition of the frequency distribution leads to the result that the sum of all context frequencies of a single word will be 1, which again allows us to treat it for all purposes as a probability density distribution.

$$\begin{aligned} \sum_i c_i(\mathrm{w}_j) &= \sum_i \frac{\|\{c_i | t_0 = \mathrm{w}_j\}\|}{\|\mathrm{w}_j\|} \\ &= \frac{1}{\|w_j\|} \sum_i \|\{c_i | t_0 = w_j\}\| \\ &= 1 \end{aligned} \tag{3.9}$$

These two representations, the context frequency of words, and the word frequency of contexts, gives us a dual model of the associative properties of the language, from which we can begin to decipher the constraints governing these associations.

Some problems

This dual representation of the language, in terms of the context frequency of words, and the word frequency of contexts, can give us a very clear picture of the interplay of lexical and syntactic constraints present in the language. By explicitly defining those sets of words participating in a contruction, and the constructions alowable for each word, we get a clear delimination of the boundaries and constraints governing their interaction.

It is, however, a grossly impractical proposal. For a corpus with a vocabulary of size V, there are $\|C_{n,m}\| = V^n + V^m$ possible (n, m)-contexts which might appear. For a language such as English, with a vocabulary in the hundreds of thousands, a given corpus would have to be immensely long to expect to see even a small fraction of the total contexts possible. Or, we would have to study incredibly short contexts of one or two words. This is the approach generally taken for keeping the number of possible observations to a sensible limit.

The later sections of this chapter will describe a different approach, however, using categories of words and contexts to reduce the number of observations to a tractable level. The structure of the model will remain the same, with contexts projecting constraints on their participating members, but the participants will instead be classes of possible words, and classes of possible constructions, rather than explicitly observed sequences of words.

3.1.5 An aside into probabilities and n-grams

Much of the current literature in statistical natural language processing is concerned with estimating the conditional probabilities of n-gram distributions, for much the same reasons as I am interested in frequency estimates of context appearances. Such estimates form a basic representation of the behavior of words, and as such, can be used for many predictive and explanatory purposes.

An n-gram, in general, is a group of n lexical items in some specified relation. Usually, though, the term n-gram is used to refer to a contiguous sequence of n items. In modelling language as a Markov transition process, the conditional dependence usually sought is the probability density of the n^{th} item w.r.t. the remaining $n-1$, as given in eqn. 3.10.

$$\ldots, t_1, t_2, \ldots, t_{n-1} : t_n \quad \Longrightarrow \quad p(t_n | t_1 - t_{n-1}) \tag{3.10}$$

This function $p()$, represents the probability of observing a particular token in position t_n after having observed exactly the tokens $t_1 - t_{n-1}$.

These probabilities are typically estimated as a function of the observed frequencies of the various events. That is, the number of observed events in which, for a specified set of $[t_1 - t_{n-1}]$, the token t_n is observed to be $\mathrm{w}_1, \mathrm{w}_2$, etc., relative to the numbers of all events is taken as the base estimate for the function $p()$ (see [Brown *et al.*, 1992a] or [Dagan *et al.*, 1994], for example).[3] This *sample frequency* is taken to be representative of a true, underlying probability of occurence, and is used to estimate these true probabilities. This is complicated though by the fact the the nature of language data almost guarantees that the sample will be incomplete (see forward to section 3.2 on sparse data).

This n-gram description (eqn. 3.10) is entirely equivalent to the word frequency distribution (eqn. 3.5) of the context $\mathrm{W}(c)$ for the $(n-1,0)$-context $c = [t_1 - t_{n-1}], t_0, []$. In a probabilistic notation, the word frequency distribution could be written as

$$\begin{aligned} \mathrm{W}(c_i) &= p(w|c_i) \\ &= [p(w = w_1|c_i), p(w = w_2|c_i), \ldots, p(w = w_V|c_i)] \end{aligned} \tag{3.11}$$

However, one should remember that $\mathrm{W}(c)$ is not a probability, nor is it intended to be, but rather it is a *sample frequency* estimate.

The reason for highlighting this difference in approach, terminology and notation stems from the belief that the underlying events being described (linguistic expressions), and especially the likelihood that a given estimated model is appropriate, are not necessarily amenable to description in probabilistic terms. A probabilistic description may also reinforce an assumption that the underlying cognitive processes may somehow have a random basis. While the behavior may certainly be modelled in terms of the frequencies of various actions, the notion that the causative mechanisms have a probabilistic nature is not one I intend to support.

Even when cognitive concerns are ignored, the fit of a model to the observed data may not be entirely evaluable in such terms, especially considering the varying foundations of probability available within the science of statistics itself.[4] Since the

[3] Check this reference carefully!!! see Maegerman's review for a sampling...

[4] See [Edwards, 1972] for a comparison of the likelihood interpretation and the probability interpretation of statistical models.

events can described statistically in terms of expected occurrence frequencies while maintaining the same results, I will attempt, as a matter of philosophy, to avoid the probabilistic terminology. Where it does appear, please remember that a frequentist approach is intended.

3.2 Sparse data

This distributional model of word comparison is fine in the ideal, but the world of real data makes it quite difficult to pursue. With a realistically large vocabulary of, say, at least V=10^5, the number of possible distinct contexts for even the very short $(2,0)$-context (trigram) model swells to $V^3 = 10^{15}$, or one *trillion* contexts! Given that current large text data resources count their words in the 100's of millions, or occasionally in the billions[5] [?], the size of the data relative to the number of possibilities guarantees that most of the 10^{15} distinct c_i's will not appear even once. And, for those that do, the count $\|\{c_i|t_0 = \mathrm{w}_j\}\|$ will be zero for almost all words. Gale and Church [Gale and Church, 1990] have observed that even as the test resources grow, the observed vocabulary size V grows at a rate greater than $\sqrt{N}$ (where N is the corpus size), which restricts even the possibility of forming a complete bigram model.[6]

This affects the possible analysis tremendously. We are already assuming that there are grammatical constraints which will rule out the use of a certain number of patterns. If a certain set of patterns aren't observed, perhaps these patterns are excluded from the language by these constraints. This is the naively optimistic conclusion; it might also happen that if we had more data, these excluded patterns would indeed appear.

[5] find the cite of ibm's paper with the 9 gig test db

[6] Check this conclusion. Do they really say that, with their $\sqrt{n}$ argument? Will i get fried for this?

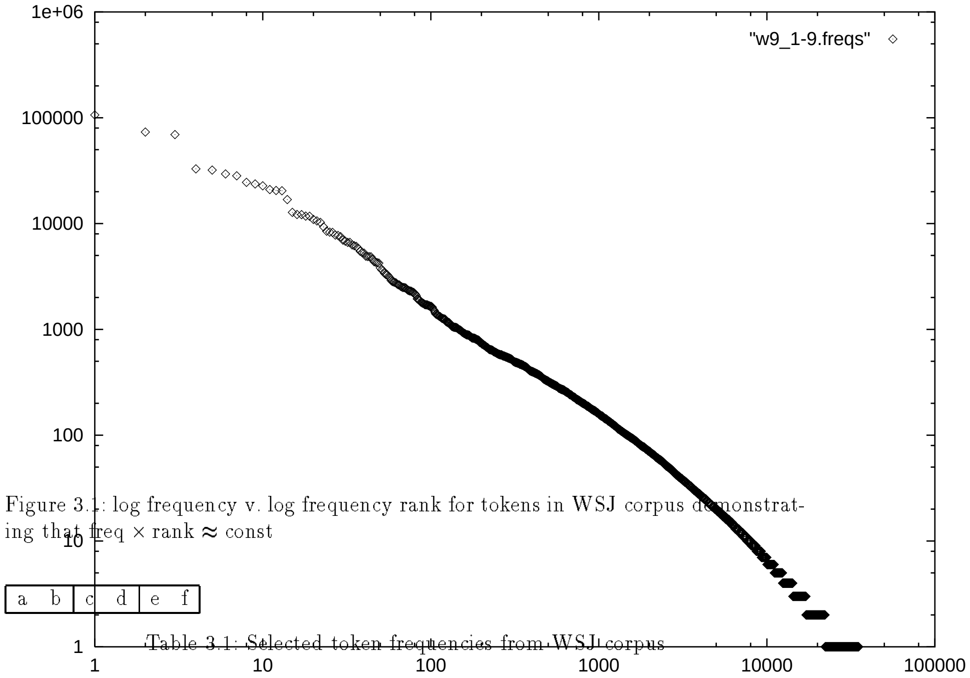


Figure 3.1: log frequency v. log frequency rank for tokens in WSJ corpus demonstrating that freq $\times$ rank $\approx$ const

a	b	c	d	e	f

Table 3.1: Selected token frequencies from WSJ corpus

Word frequencies

The problem is compounded further by the fact that the frequency distribution of single words generally follows a form known as Zipf's Law [Zipf, 1972, pp. 25 ff.], which equates word frequency with the inverse of the frequency rank through the formula $f(w) = K/r(w)$, where $f(w)$ is the number of occurrences of the word w, and $r(w)$ is the *rank* of that word in terms of frequency, with the most frequent word having rank = 1, and the least frequent word rank = V.

This relation means common words are far more ubiquitous than uncommon words. Figure 3.1 and table 3.1 demonstrate this for the news corpus used in the experiments of chapters 4 and 5. Although it is somewhat difficult to see in the log-log plot, low frequency words make up a large proportion of the total data. Singleton words (freq=1) comprise 12985 of the 35187 distinct tokens in this sample, acounting for 37% of the total vocabulary.

Context frequencies

This form of word distribution within the language also conspires against us when we try to estimate context frequencies. Even if we ignore the structural dependence among the symbols of a context, the expected frequency of contexts containing even one, much less two or more of these low frequency symbols will be astronomically low.

Consider for a moment a sample in contradiction to Zipf's distribution, and in contradiction to known facts about interword dependence: an order-0 sample in which word frequencies are uniform and independent. With a vocabulary size V and a sample size N, each word then has, by definition, frequency N/V. The probability that any given token in the sample is a particular word is $1/V$. The chance then that a particular context of length k occurs as a given subsequence of length k is V^{-k}. Since there are $N - k \approx N$ such subsequences in the sample, the appearance probability $a(c_i)$ that a given context appears somewhere in the entire sample is $1 - (1 - V^{-k})^N$ (the complement of the chance that it does *not* appear). The second term, $(1 - V^{-k})^N$, is exactly $\sum_{j=0}^{N} \binom{N}{j} V^{-jk}$. For large $V^k \gg N$, we can estimate this term as $(1 - V^{-k})^N \approx 1 - NV^{-k}$. So for a context c_i of length k, $a(c_i) \approx NV^{-k}$.

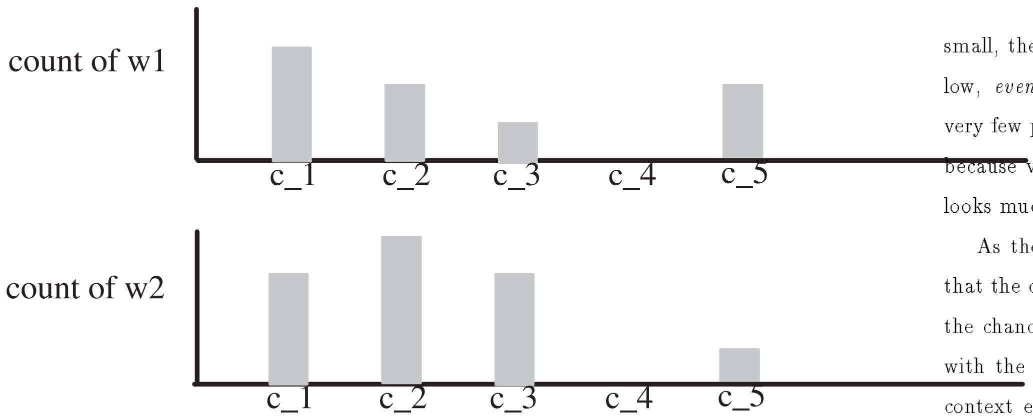
If we want to verify, directly from our observations, that a given c_i is in the language, or to conclude that it is excluded from the language due to structural constraints, we need a sample large enough to make the probability of this null hypothesis reasonable. This estimate implies that for a $1/10$ chance of seeing this particular context we need a sample of length at least $N = V^k/10$. This is already unreasonable for the case of English trigrams, where $V^k > 10^{15}$. The situation becomes exponentially worse as we choose to look at longer contexts.

Further, since the language is Zipf distributed, most symbols have very low frequencies, and any chance of finding a context composed of low frequency symbols requires astronomical amounts of data. If the interesting symbols have, say, as much as half the average frequency (already a high estimate) then we need a sample length at least $N = 2^k V^k/10$, which is 2^k times larger than needed previously. This is prohibitive, to say the least. Since the Zipf distribution implies that most symbols have even lower frequency, we need even a larger corpus, and it can be seen that acquiring enough data simply becomes impossible.

3.3 Comparing distributions

The most serious problem with such sparse data is that, if we have no basis for comparison other than exact context match, there will be so few occurrences of the same context appearing more than once that we will have no basis for comparison at all. Earlier, I suggested using the context frequency distribution as a mechanism for comparing the grammatical properties of words. The idea presented was that by considering the contexts in which two words appeared, one could gain an understanding of the associative constraints common to both by observing the commonalities within the contexts — the similarities and differences of their projected constraints would be apparent in the common features of their context sets. The situation described could be concieved as shown in figure 3.2, where the common contexts c_1 and c_3 indicate a shared grammatical constraint, whereas the appearence of only one word with c_4 helps determine which context properties of c_1 and c_3 should *not* be considered in a comparison.

However, the incredibly low probability of contexts has a serious consequence on this proposal. Because the chances that any particular context will appear are so



histogram with word appearance over 5 or 6 contexts, reasonable overlap.

Figure 3.2: Comparing two words across a limited number of high-probability contexts.

'spikey' histograms over many contexts, with maybe one spike overlap

Figure 3.3: Comparing two words across an enormous number of low-probability contexts.

Same spikey histogram, but with wide average-for-a-class bars over the spikes.

Figure 3.4: Comparing two words across a large number of low-probability contexts, smoothed using categorization.

small, the chances that two words will appear in an identical context are incredibly low, *even if the words project very similar constraints!* In this case, there will be very few pairs of words that one can even compare through their context frequencies, because very few pairs will appear in identical contexts. The diagram for this case looks much more like figure 3.3 than it does figure 3.2.

As the last section made clear, limited data and sparse distributions guarantee that the count $\|\{c_i|t_0 = \mathrm{w}_j\}\|$ will be zero for most words. As a further complication, the chance that a given context will be observed in corpus diminishes exponentially with the length of the context, meaning that if we were to represent each possible context explicitly, most counts will simply be zero. This implies that any system, human or machine, attempting to acquire a notion of syntactic constraints based on structural similarity will have to use features less constrained than exact structural match.

3.3.1 Smoothing

One approach to making these context distributions more comparable is to *smooth* the distributions. Since the data has been gathered through a discrete process, with a context appearing either an integer number of times or not at all, the collected statistics will suffer from quantization effects. If a low probability event appears even once, its frequency will be estimated many times too high. Likewise, if a moderate frequency event appears even one too few times within the particular sample of text, Non-sparse set of points, with a couple missing, and a couple off peaks, with a dotted line fitted to the peaks.

Figure 3.5: Interpolating missing values smooths the curve.

its frequency will be grossly underestimated. Because of the size difference between the enormous space of possible events and the small available samples of contexts, low frequency events will be the most misrepresented within the sample. An event that does not appear in the sample could easily have a non-zero chance of occurring. Smoothing operations re-estimate the sampled frequencies in order to reduce the effects for this quantization.

Smoothing data from a continuous source, such as measurements from a physical system, is made simpler by the assumptions that can be made about the underlying form of the data. Assumptions about the metric space in which the events occur, continuity assumptions, and limited derivatives resulting from the physical limits of the device being measure contribute to simplifying the task of estimating missing values, identifying spurious measurements, and drawing relations between the observations. The situation is often as in figure 3.5 in which missing data can be filled by interpolation, and erroneous data can be detected and normalized using techniques such as local averaging. The end result is that the data can be fitted to the form of a smooth curve, because there is a smooth model of the underlying phenomenon.

Unfortunately, a more accurate picture for linguistic data is a shown in figure 3.6, where most values are missing. These missing values are grossly underdetermined without an model of the underlying behavior of the language. Each point must be treated independently, eliminating local averaging and interpolation as applicable methods.

For an example of these extremely sparse condidtions, Brown et al. [Brown *et al.*, 1992a] report that for a particular 3-gram model, with a corpus of 3.7×10^8 words, only 7.5×10^7 of 1.8×10^{16} possible 3-grams appeared at all, for a data rate of 4.3×10^{-9}.

sparse set of points, with large gaps filled with large question marks, small sections of curve fitted over the few peaks.

Figure 3.6: Smoothing a sparse set. missing values are underdetermined.

And, even of those that did appear, 71% appeared *only once*.

Non-linguistic smoothing estimates

We have seen that if each event to be estimated is considered independently, enormous quantities of data are required in order to obtain reliable frequency estimates. One needs in this case a sample large enough so that there is a high confidence in the observed frequencies, even for low-frequency events.

Without having a continuous model with exploitable local properties, as in a physical system, one must use other assumptions about the theoretical shape of the phenomena in order to smooth the data. There are some smoothing techniques which really on very broad assumptions about the structure of the data in order to re-estimate observed frequencies. Methods such as the Good-Turing estimate [?] and the *deleted* estimate [?] assign estimated frequencies to each unobserved event based on the relation between the number of possible events and the number which were observed.

The deleted estimate relies only on the assumption that the same stationary process underlies two separate language samples. The Good Turing estimate [Good, 1953] uses this and the assumption that the count of each symbol is binomially distributed. Both of these methods treat all events observed exactly k times (including k=0) as a class, and assigns an estimated frequency uniformly to the members of the class. Unfortunately, this means that the frequency predicted for all unobserved events, including those which are not part of the language, is the same as that predicted for allowable events whose frequency is low enough that they do not appear in the sample.

Church and Gale [?] introduce *enhanced* Good Turing and deleted estimates, using the additional information of the distribution of single words (or *unigrams*). This information is used to categorize the possible events according to the expected frequency as if the constituent words were independent, in order to improve the resolution of the frequency assignment. Although they report a considerable improvement in the predicted frequencies under this method, the technique still uses no linguistic information and cannot distinguish between meaningful and non-meaningful arrangements of symbols. A non-member string composed of high-frequency words that was not observed (for good reason!) will still be assigned an estimated frequency higher than an unobserved, but allowable event that was not observed due only to low frequency.

There are many other approaches to improving the resolution of frequency assignment. Jelinek and Mercer [?] propose an *interpolated estimation* technique for n-gram frequency estimates (eqn. 3.12).

$$\begin{aligned} f(\mathrm{w}_n|w_1-\mathrm{w}_{n-1}) = & + \lambda_n(\mathrm{w}_1-\mathrm{w}_{n-1})f_n(\mathrm{w}_1-\mathrm{w}_{n-1}) \\ & + \lambda_{n-1}(\mathrm{w}_2-\mathrm{w}_{n-1})f_{n-1}(\mathrm{w}_2-\mathrm{w}_{n-1}) \\ & + \ldots \\ & + \lambda_1 f_1(\mathrm{w}_n) \end{aligned} \tag{3.12}$$

In this method, if the higher-order predictors ($f_n()$) are deemed unreliable or uninformed due to a lack of sufficient data, they are given a low weight (λ), and the brunt of the prediction is shifted to the lower order estimates. This decision is often referred to as 'backing off' to a lower order model. One can also view this weighting from the opposite direction, and treat the model as a low order estimate, which is refined by the inclusion of higher-order statistics when there is sufficient evidence to judge them reliable.

The major drawback to estimates of this kind, however, is that they completely

ignore linguistic abstractions of the data. They treat the data as a stochastic stream of symbols, for which the redundancies and repetitive structures can be modeled in terms of the predictive capacities of one or two symbols. This approach ignores the view of language as a *meaningful* stream of symbols, where the interrelations between symbols are governed, through a process of interpretation, by their correspondence to semantic structures.

Treating language as a stochastic stream is not entirely a step backwards, though. It provides a larger view of language than we can garner from introspecting on sets of examples, or investigating correlations between small sets of occurrences. But, because we know so much more about language, its limitted stochastic properties can serve only as a groundwork for more complete and acurate models.

3.3.2 Linguistically informed methods

Adding linguistic information to a model can improve its accuracy. In particular, a linguistically motivated acount of word behavior can be used to beter estimate the frequencies of unobserved events. A variety of linguistic generalizations can be used to improve these interpolation methods, and model unseen events by analogy with the observed behavior. Dagan and others [?] have presented methods which take into account similarities between words in order to predict word use. In order to better model a word w_i's expected behavior, they average the distributions of 'similar' words to create a more robust, composite estimate of the distribution for w_i. Similarity in this case was arrived at using an information-theoretic measure, the Kullback-Liebler distance. The KL distance measures the degree to which one distribution predicts another, so that a word whose distribution of behavior closely matches the target word's observed distribution is used to round-out the target's expected distribution of behavior.

The assumption behind this is of course that words whose behavior closely approximates one another's are filling the same functional role in the language. Thus, an appearance by one in a situation implies that the other is also licensed there.

These similarity judgements can be computed only where there is some reasonable overlap of behavior. If the data is too sparsely distributed, then the chances that two words will have any behavior in common is also low. The KL distance is computed over a discrete distribution through a comparison of probabilities in each cell. If two words share little or no behavior exactly, then their similarity will be very low.

Because Dagan et al. use bigram data, their similarity estimates are robust enough to be practical. A larger context, or longer n-gram, would reduce the data density so much, though, that I do not believe these methods would be extensible past the bigram.

3.3.3 Classification methods

The most obvious abstract linguistic extensions to the stochastic models can be drawn from the existance of constrained, regular syntactic structure. By noting that individual words do not behave completely idiosyncratically, but instead follow patterns of behavior similar to others, we immediately arrive at the idea of *word classes*.

If we observe the interactions among this reduced set of classes, rather than among the numerous individual words, the available data becomes much more complete. Estimation of the occurrences and interactions among the classes becomes tractable. From a model of word classes, the behavior of individual words could be estimated according to their membership in particular classes.

Many approaches of this type have been developed. [Brown *et al.*, 1992a] presents a tri-gram Markov model

— finch

— schutze.

Limitations of fixed word classes

These class-based methods suffer a serious drawback, however, taking the linguistic generalization too far. In assigning a word to only a single class, and then using the class behavior as a representation for the word's behavior, they imply that each member word of the class is completely substitutible for any other. The class representation may indeed be an improvement over individual word statistics, but this overgeneralization is perhaps too severe. In truth, each word may have a variety of aspects of behavior, any one of which it shares with a number of other words. This is the same as saying that it may assume a number of differenct functional roles within the language, each operating under its own set of constraints. But to say that all the roles and constraints associated with a word are to be found shared by all other members of the class, and to the same degree, is an assumption that we need not make.

This overzealous grouping can be avoided by adopting a class representation in which each word can have membership in more than one class. In [?], Rooth presents a class-based model of noun-verb selectional types. Each selection type in this model carries both a probability distribution of verbs and one of nouns, indicating the words most likely to appear in interactions of this type. Each type carries distinct distributions. Furthermore, each type is assigned an occurence probability (eqn. 3.14).

$$\begin{array}{rll} \text{verb probability} & p_v^\tau, & \sum_v p_v^\tau = 1 \\ \text{noun probability} & p_n^\tau, & \sum_n p_n^\tau = 1 \\ \text{type probability} & p_\tau, & \sum_\tau p_\tau = 1 \end{array} \tag{3.13}$$

These distinct distributions allow us a variety of ways to view the data. The p_v^τ distribution gives us the chances of all verbs with respect to a given type of N-V interaction. One can also view this as the chances of a verb given a certain expectation of the noun. The type τ effectively defines a fuzzy class of nouns, and this distribution p_v^τ tells us the expected relationship of verbs to this class. The distribution p_n^τ is the dual of this, from the point of view of the noun.

Of course, this model deals only with the N-V relationship, and relies on having input pre-tagged in that respect. Modelling only this limitted interdependence also eliminates much of the syntactic diversity and ambiguity present in language. English words are remarkably flexible in their usage, and the same term can appear in verbal, nomimal and modifier roles, with perhaps multiple sense interpretations in each. Dissecting and categorizing this complex web of usage is accordingly more difficult.

3.4 Associative class description

[repeat conclusions from prev. chapter] Conclusion:

1) for given word, the words appearing locally with it will appear only in certain configurations, based on their classes.

2) conversely, words of the same class will have similar configurations that appear with them.

3.4.1 Purpose

The intent behind such a division of words into classes comes, as it should for all classifications, from the *descriptive* and thus the *predictive* power that any representation brings to it's subject.

The descriptive power of such a representation comes from the implicit description of the syntactic properties of the langague as reflected by the features measured by $s(a, b)$.

Reduced representation By glossing these features with an abstract set of categories, the classification gives a more managable representation than the original data, one of the [??] goals of scientific investigation.

Captures regularities The classification captures the regularities of the language w.r.t the features used in $s(a, b)$, making this higher order structure available for investigation.

The predictive power of the representation stems from the regularities expressed within it. An abstraction of this sort seeks to capture the behavior of the system, and in as much as the system has a regular behavior, to predict it's future behavior from the past. The syntactic classes serve to constrain one's expectations on the behavior of their members.

Class based stuff cut from old ch. 4

3.4.2 n-grams and n-contexts

The *n-gram* representation has long been used by the speech recognition community for phoneme and word sequence modeling, and has been adopted more recently for more general linguistic studies. There is a direct correspondence between the (n, m)-context and the $n-$gram representations.

An n-gram is an n-item subsequence of a corpus. Often, statistics on n-grams are used to estimate occurrence probabilities of words or sequences of words in language. Usually, these statistics take the form of marginal distributions on the word value of the first or last item in the sequence.

For instance a left tri-gram (3-gram) model would use all 3-item subsequences of a given body of text. This is equivalent to a $(3, 0)$-context. The distribution of the first two items of each 3-gram can then be used to estimate the probability of the third:

$$\{[t_{i-2}, t_{i-1}, t_i]\}_{i>2} \implies \Pr(t_i = w_j | t_{i-2}t_{i-1} = w_a w_b) \tag{3.14}$$

These probabilities can then be interpreted as transition probabilities on a hidden Markov model, indicating the probability of encountering a word w_j after having seen the pair $w_a w_b$.

Note that this Markov model can be seen as a classification of of words into categories corresponding to all pairs $w_a w_b$. Thus for a vocabulary of size V, we have V^2 categories named by the pairs $\omega_{a,b} \leftrightarrow w_a w_b$, $1 \leq a, b \leq V$, and category membership defined as the estimated probability of occurrence.

$$\omega_{a,b}(w_j) = \Pr(w_j | w_a w_b) \tag{3.15}$$

Note that this model requires the estimation of V^3 parameters ,V^2 probabilities for each of V words (V^n parameters in general for an n-gram model), making the process of estimation rather difficult, and making the size of the model rather cumbersome.

One could reverse the conditioning in interpreting the statistics, calculating instead $\Pr(t_{i-2} t_{i-1} = w_a w_b | t_i = w_j)$. This has an classifier interpretation as V categories with V^2 members each, with membership defined as

$$w_j(w_a w_b) = \Pr(w_a w_b | w_j) \tag{3.16}$$

Both of these class interpretations use the maximum number of classes possible, and thus do not offer any any of the advantages

3.5 Word-based classification

One way of dividing the lexis of $\mathcal{L}$ into classes $\omega_1, \omega_2, \ldots, \omega_n \in \Omega$, is to assign words to classes so that each word of a class appears in configurations that are more similar those corresponding to words in its own class than they are to those from other classes. We can notate this classification using a *partition function*, π, where $\pi(w_i) = \omega_j$ assigns each word $w_i \in \mathcal{L}$ to one class $\omega_j \in \Omega$.

In order to quantify the notion of similarity between configurations, a *similarity measure* can be used to capture those features of the configuration that we wish to compare. In the case where we compare the *total* usage of words (all the appearances taken together), we can notate the similarity measure as $s(C(a), C(b))$, where a and b are words of $\mathcal{L}$, and $C(a)$ is some gneralization of all the contexts in which a has appeared.

The similarity within and between classes, s_w, s_b, respectively, can be defined with respect to a partition π, a generalization of contexts, $C(a)$, a similarity function between contexts $s\,(C(a), C(b))$, and an averaging or generalizing function $\sum'$. This averaging function is necessary to conteract differences in size among the classes, so that large classes do not accrue greater total measures than small classes.

$$s_w^\pi(a \in \omega_j) \stackrel{\text{def}}{=} \sum_{b \in \omega_j}{}' s\,(C(a), C(b)) \tag{3.17}$$

$$s_b^\pi(a \in \omega_j) \stackrel{\text{def}}{=} \sum_{b \in \omega_i \neq \omega_j}{}' s\,(C(a), C(b)) \tag{3.18}$$

The separation of word into classes can now be stated as a constraint on the values of the between and within class similarities. For a division into n classes, $\omega_1 - \omega_n \in \mathcal{C}$, we wish to find a π which assigns each word $w_i \in \mathcal{L}$ into a class ω_j such that

$$s_w(a) > s_b(a) \qquad \forall a \in \mathcal{L} \tag{3.19}$$

One may even want to find a π which maximizes the distinction between the classes. That task can be represented as finding $\pi_{\max}$ where

$$D(\pi_{\max}) = \max_{\pi} \sum_{a \in \mathcal{L}} s_w^\pi(a) - s_b^\pi(a) \tag{3.20}$$

Note that this is not the only possible measure of class separability. In general, one would want to find the maximum over some function of within and between class similarity measures:

$$D(\pi_{\max}) = \max_{\pi} \sum_{a \,\epsilon\, \mathcal{L}} f(s^{\pi}_{w}(a), s^{\pi}_{b}(a)) \tag{3.21}$$

3.5.1 Building word-level descriptions

Each context c_i represents a single use of the symbol appearing at t_i in context. By classifying the contexts, the words themselves have also been categorized in terms of the contexts in which they are used.

This relation between the context-based partition and the individual words used has two natural expressions.

The class distribution

The inverse relation between categories and words is the *class distribution* assigned to a given word.

Since each occurrence of a word is tied to a configuration class through its context, we can describe the total usage of a word in terms of the distribution of categories that it takes.

Analogously to the word distribution of a class, we can define the proportion of a given class, ω_j, with respect to the total of all classes assigned to a given word, w_k. We can define this as the ratio of the size of the subset of all assignments of the word to the given category to the total number of appearences of the word. While this total is really $\|\{t_i \,\epsilon\, t \mid t_i = w_k\}\|$, we can use the obvious shorthand $\|w_k\|$.

So the subset of assignments can be defined:

$$w_k/\omega_j \stackrel{\text{def}}{=} \{c_i \,\epsilon\, \omega_j \mid t_i = w_k\} \tag{3.22}$$

and the proportion of such assignments to the total appearances

$$\omega_j(w_k) \stackrel{\text{def}}{=} \frac{\|w_k/\omega_j\|}{\|w_k\|} \tag{3.23}$$

giving us the class distribution of a given word, $\Omega(w_k)$

$$\Omega(w_k) \stackrel{\text{def}}{=} [\omega_1(w_k), \omega_2(w_k), \ldots, \omega_K(w_k)]^T \tag{3.24}$$

3.6 Relating contexts to words

Word-based classifications use mixtures of context-based The context-based classification is a finer-grained

3.6.1 Related work

Brent.

Simmons and Yu.

??

3.6.2 Related work

- n-gram models: — implicit ad-hoc classes when used predictively, as in speech recognition.
- n-gram class models: — *a la* Brown's CL article. Syntactic as well as semantic classes.
- Context-summation clustering: Schütze; verb clustering and northeastern bio-literature work from 1993 SIGLEX.

  Finch; Finch and Chater.

  John Hughes (thesis)

  Fu and Lu.

- Syn-set stuff: George Miller.

  Sparck-Jones.

  Other IR stuff.

3.7 Limitations

Limited data — probably not enough occurences of all words. Perhaps not even for all categories.

Sparse data — in the context of a given word, probably not enough overlap with words found in other contexts to get truly good results.

limiting computational factors:

- context size: number of words to keep track of for each occurrence. kills off studies of the occasional long distance interaction.
- number of occurrences studied. Some phenomena will certainly be missed.

3.8 Summary

reiterate why classification is important

- data reduction. Raw data too much to grok. Classification gives a condensed version that retains important qualities of original.
- analytic reasons. It's easier to state properties over classes than over raw data.
- predictive power. An abstraction allows predictions to be made in terms of expected data.

reiterate connectons to standard POS analyis, subcategorization, feature-based categories, etc.

3.9 Crap from elsewhere

What, however, are the appropriate measures to use in assessing these similarities?

[What crap — I launch right into a discussion of crisp v. fuzzy categories.

should jsut try the obvious – direct matching of the distributions, and then explain that because of sparse data, you won't actually see much (if any) overlap between ANY two tokens.

Consider two words, w_1, w_2, appearing in contexts c_1, c_2, respectively.

Using crisp (Boolean) categories, in which each context c_i is assigned to only one category ω_j (*i.e.* $c_i \in \omega_j$), if it happens that $c_1 \in \omega_j$ and $c_2 \in \omega_j$, we can say only that w_1 and w_2 share this common behavior, and that we might expect either word to appear in any of the other contexts belonging to the same category $\{c_k | c_k \in \omega_j\}$.

However, using a context measure $d(c_1, c_2)$, we can make the broader claim that w_1, which was observed in context c_1, can be expected to appear in a different context c_2 as determined by the context measure $d(c_1, c_2)$. The exact expectation of this event would also depend on the relative frequency of w_1 in c_1:

$$E\left(\mathrm{w}_1(c_2)\right) \sim f\left(d(c_1, c_2), \mathrm{w}_1(c_1)\right) \tag{3.25}$$

In general, if we wanted to know the complete word frequency distribution for c_2, $\mathrm{W}(c_2)$ (as in speech recognition), we could sum the distributions of all other contexts c_i, weighted by their similarity to c_2:

$$E\left(\mathrm{W}(c_2)\right) = \sum_{c_i} g\left(d(c_i, c_2), \mathrm{W}(c_i)\right) \tag{3.26}$$

Chapter 4

Quantifying similarity

In the previous chapter, I outlined a method of *local distributional smoothing* that I argued better matched the problem of word class formation and occurrence estimation than standard smoothing methods, due to the severity of the sparse data effects. As mentioned there, this sort of smoothing operation requires a measure of *locality* within the sample space, which in this case is a space of short context sequences. Such a similarity measure defines an implicit structure to the space of possible data, allowing one to represent the frequency of various phenomena as distributions throughout the structured space, rather than as point-valued phenomena defined only on the observed data.

The first part of this chapter revolves around the issues involved in defining a measure which can be used to structure the data space. Section §4.1 discusses various sequence measures that can be used to structure the space of possible contexts, and defines a local region within the data space that is required for the local smoothing. Section §?? gives some quantitative values of these measures over real language data, comparing the various choices of string measure. The results from real data are also compared to those from data generated from the first-order word frequency distribution, and deviations between the actual and extimated results are discussed in terms

of the additional, higher-order structure present in the sampled language data.

4.1 Sequence measures

Central to the discussion of the last chapter was the notion of a similarity metric $s(c_i, c_j)$. This metric captures the relevant linguistic structure of sequences of symbols, and tells us when two sequences have like structure. We then use this metric to categorize all like sequences into a class, in effect averaging their properties into the class description. By categorizing all possible sequences into some managable set of context classes, we reduce the complexity of the representation and create a model based on these context classes.

A linguistically meaningful reduction of the space of possible contexts would have to accomplish this averaging while preserving the scope of linguistic inference we wish to derive from the model. Such a reduction would preserve our intuitions about paradigmatic substitution classes, namely, that they represent lexical items that are interchangable under limited conditions. In order to collapse similar contexts into a local region, we must make some allowance for disagreements in context comparison. Replacing an element of the context with a close synonym, for example, should not greatly affect the comparison. Neither should replacing that one element with a synonymous multi-word phrase.

One could imagine many possible means of comparing two sequences of tokens. We are searching for a comparison, however, that preserves the specific sort of meaningful *linguistic* similarity discussed in ch. 2. Such a measure would incorporate information relating both the identity of the tokens, and their configuration.

A measure of paradigmatic similarity would provide a starting point for such comparisons between contexts, providing the basic judgement used to measure the total disagreement between contexts. However, we start with no model of language,[1] and the data we have contain no measure of synonymy Such a measure is, in fact, part of what we are hoping to derive from this study. The data as it is given is represented only as a sequence of tokens. Because we are not starting with a model of language, these tokens have *nominal values* only. That is, they can only be compared in the sense that any two tokens either have identical values, or not, giving us a match/not-match binary comparison as a starting point for measuring context similarity. By counting the numbers of matches/disagreements between the pairs of tokens in two contexts, we can measure the total similarity of the contexts.

In order to incorporate a linguistic sense of similarity in such a comparison, we will need to organize the way in which tokens are compared between the two contexts under consideration. There are many ways of choosing the possible pairs to be compared.

4.1.1 Set measures

One simple way of making this comparison is to ignore the sequential aspects of the contexts, and treat each as a simple set of nominal values. The most basic comparison that can be made of two sets with nominal features is a count of the features shared between the two. Symbolically, if we treat the two sequences A and B as *sets of features* drawn from some feature space $\{f_1, f_2, \ldots\}$, this count is the cardinality of their intersection, $S(A,B) \stackrel{\text{def}}{=} \|A \cup B\|$. In using this measure for comparing two lexical contexts, we can treat lexical tokens as the features from which sets are drawn, and treat the contexts as simple collections of tokens. The measure S is then the number of tokens in common between the two contexts.

[1] It might be thought that a simple model of morphology or part of speech could be used to some advantage here. That has its own difficulties, in judging the appropriateness of such a model for the task as hand, and further, of judging the numerical relationships of similarity between morphological variants or parts of speech.

This measure has number of drawbacks in terms of its linguistic appropriateness. First, and perhaps most importantly, the measure S does not take into account any ordering or configurational information that may be present in the lexical environment. The contexts are treated simply as bags of words, and any lexical constraints represented by their ordered relation to one another is completely ignored. Secondly, S is a very discrete measure, taking on only $n+m+1$ possible values in the comparison of two (n,m)-contexts (the intersection can have zero to $n+m$ elements). Given the large domain of possible contexts (see §3.2), this range of possible comparison values seems incredibly sparse. Lastly, the measure S also doesn't consider the effects of the multiple occurrences of elements with the two sets. In the extreme, if A and B were identical, but each composed only of a repeated occurence of a single token, say, x, then the measure would be $S(A,B) = 1$ no matter what their length, the same value that would be assigned two contexts which had no repeating values, but had only one element in common. At another extreme, two identical contexts of length n with no repeated tokens $A = B = [x_1, x_2, \ldots, x_n]$ would have a measure $S(A,B) = n$.

The Jaccard measure

Another basic measure for comparing item based on nominal features, which avoids this last drawback to the set measure S is the Jaccard measure, which is a comparison between the number of elements shared by two objects, and the total number of elements in their union [?, p. 89].[2] The measure is simply the ratio of overlapping elements to total elements:

$$J(A,B) \stackrel{\text{def}}{=} \frac{\|A \cup B\|}{\|A \cap B\|} \tag{4.1}$$

In using this measure for comparing two lexical contexts, we again treat tokens as the features from which sets are drawn, and the contexts, which are sequences

[2] or maybe it's anderberg. need to check. Plus look in everitt.

of tokens, as simple collections of features. Thus, the Jaccard measure between two contexts such as c_1 and c_2, below

$$c_1 = [a\ b\ p\ q] \qquad c_2 = [c\ b\ r\ q] \tag{4.2}$$

is computed as:

$$J(c_1, c_2) = \frac{\|\{a,b,p,q\} \cup \{b,c,q,r\}\|}{\|\{a,b,p,q\} \cap \{b,c,q,r\}\|} = \frac{\|\{b,q\}\|}{\|\{a,b,c,p,q,r\}\|} = \frac{2}{6} \tag{4.3}$$

By using the cardinality of the union as the denominator in the ratio, the Jaccard measure normalizes the size of the intersection set according to the intrinsic 'size' of the two sets combined, and avoids the variable results provided by S. This normalization also has the useful property of restricting the value of the measure to the range $[0, 1]$

While the Jaccard measure deals well with a variable number of distinct tokens within each context, it still suffers from the other problems which make the set measure unsuitable for linguistic analysis: it does not take configuration into account, and has a very limitted range of possible values when used to compare short contexts.

The Jaccard measure and variants have been used to some good effect in problems in information retrieval [?, ?].[3] In these applications, though, the sets being used are token representations of entire documents, containing a much wider distribution of tokens than relatively short contexts. Because of the size of the token set in the document representation, the problem of few output values does not exist for these applications. Because the Jaccard measure has been used effectively in these IR applications, the experiments in §4.3 show Jaccard results as a baseline.

Distance v. similarity

The Jaccard measure as given is a measure of *similarity*, with higher values indicating contexts that are more alike. In order to make the values comparable to those

[3]find cites fro greffenstette's new book, and KSJ's ir work.

of the other measures presented, the experiments use the *distance* measure $J' \stackrel{\text{def}}{=} 1 - J$, for which high value indicate dissimilar contexts, and a value of 0 is given to identical objects.

4.1.2 Aligned pairwise comparisons

One method of incorporating the order implicit in each sequence of token elements is through an *aligned pairwise* comparison, which only considers the similarity of elements occupying the same position in each sequence. By making this limitation on the pairs which are evaluated, the comparison takes into account the critical aspect of the configuration of symbols within each context.

We define a pairwise aligned comparison measure for two contexts, $A = [a_1, \ldots, a_n]$, $B = [b_1, \ldots, b_n]$, as the sum of pairwise disagreements between tokens from identical positions. As with the Jaccard measure, we normalize by the length of the context sequence.

$$P(A,B) = \frac{1}{n}\sum_{i=1}^{n} d_i, \quad \text{where } d_i = \begin{cases} 0 & \text{if } a_i = b_i \\ 1 & \text{if } a_i \neq b_i \end{cases} \tag{4.4}$$

The normalizing factor $1/n$ serves to restrict the values of P to the range $[0, 1]$. This measure could easily have been defined in a complimentary fashion, so that P' would have $d_1 = 1$ for equal tokens and 0 otherwise. The given method was again chosen so that it was a measure of distance between contexts, with a value of 0 for identical contexts, rather than a similarity measure.

For the two contexts given above (eqn. 4.2), the pairwise comparison measure P is computed

i	1	2	3	4
c_1	a	b	p	q
c_2	c	b	r	q
d_i	1	0	1	0

$$P(c_1, c_2) = \frac{1}{4}(1 + 0 + 1 + 0) = 1/2 \tag{4.5}$$

The measure P provides a degree of improvement over set measures such as Jaccard. By limitting the comparisons only to aligned items, P manages to capture at least the linear ordered structure of the contexts under comparison, which is completely ignored in the measure J. However, it does this in the crudest possible way, which ignores many of the obvious properties of real linguistic data. For example, we might like to think that the simple noun phrases 'the boy' and 'the tall boy' are very similar, and should not be regarded very differently in a formal analysis of a larger expression in which they might be embedded. One has a modifier, the other does not, but our intuition is that they will behave quite similarly in use. However, because the pairwise metric P forces a strict alignment of tokens, it will indicate that these two phrases are extremely different.

Embedding the phrases as the subject of the VP 'went to the store', we can see explicitly why this is the case. Below (4.6) are listed the the $(0, 4)$-contexts of the token t_0 ='the' for three phrases, which we would think should be judged as fairly similar.

$(0, 4)$-context

c_1	the	boy	went	to	the	store	
c_2	the	tall	boy	went	to	the	store
c_3	the	girl	went	to	the	store	

(4.6)

One notices immediately that the additional token 'tall' brings the second sequence out of alignment. The pairwise comparisons will in this case fail to capture the similarity of the the remainder of the sequence, failing to account for this shift in alignment.

The measure between the first two is large, $P(c_1, c_2) = 1$, while $P(c_1, c_3) = 1/4$. The alignment forced by the simple pairwise comparisons is not in agreement with the linguistic intuition that the two variants, c_2 and c_3, do not differ so wildly. Certainly not so much that the pair (c_1, c_2) should be considered maximally distance, while the pair (c_1, c_3) is measured as relatively similar.

Related use of pairwise comparisons

Despite its drawbacks, this measure has been used to reasonable effect in some studies of the syntactic relatedness of words. This aligned pairwise comparison is very close to the measures underlying work such as that represented by [Schütze, 1993, Finch and Chater, 1992, Hughs, 1994, Futrelle and Gauch, 1993]. Each of these approaches aligns a set of contexts appearing with a single word, w, and tallies the occurrences of tokens within each aligned position. In the diagram 4.7, the $f(t_i)$ represent these tallied frequency distributions for each aligned position. These tallies represent the average identity of the position t_i for the collection of separate appearances of w.

$$\begin{array}{cccc|c|cccc}
\ldots & t^1_{-3} & t^1_{-2} & t^1_{-1} & w & t^1_1 & t^1_2 & t^1_3 & \ldots \\
\ldots & t^2_{-3} & t^2_{-2} & t^2_{-1} & w & t^2_1 & t^2_2 & t^2_3 & \ldots \\
\ldots & t^3_{-3} & t^3_{-2} & t^3_{-1} & w & t^3_1 & t^3_2 & t^3_3 & \ldots \\
\hline
\ldots & f(t_{-3}) & f(t_{-2}) & f(t_{-1}) & w & f(t_1) & f(t_2) & f(t_3) & \ldots
\end{array} \tag{4.7}$$

A set of $f(w_k)$ is composed for each word w_k in the study, and a comparison of these tallies is used to evaluate the syntactic similarity of the words. Since this representation of the use of a word is very large, requiring a large distribution for each of several context positions, a severe amount of data reduction is often performed [Schütze, 1993] before the comparison between words is actually performed. However, the comparisons between words amount to a measure of the similarity of distributions in each position, which is essentially a weighted count of the occurrences of tokens in that position for both words.

This type of analysis first groups all contexts occurring with a given central word w, and seeks to characterize the similarities among the averaged formal properties of the contexts. The central word is treated as an *independent* variable, and the syntactic environment is studied in terms of dependencies on the choice of word. In attacking the problem this way, these studies loose the rich, polymorphic behavior of individual words, conflating the constraints and dependencies that may operate different on separate modes of use, reulting in words treated as atomic entities with monolithic behavior. Because the lexical invironment, containing many words, is more complex than the single central element, studying the realtionship in this direction will necesarily lose information.

In the present work, I am attempting to discover and model the structure of the syntactic environment, which is more variable than the choice of central word, and use that to infer properties and features of the central element, treating it as the **de**pendent variable. This preserves the information relating a single word to its many possible uses, and

4.1.3 Edit Distance

The accepted method for overcoming the difficulties of fixed alignments in pairwise comparison of sequences is to allow for a flexible alignment of tokens between the two contexts. This is often accomplished using the *edit distance* [Levenshtein,], a measure based on relating pairs of tokens between the two sequences in a structured way, but not necessarily in strict seqential alignment. Insertions and deletions of intervene-ing, uncompared tokens are also allowed. Straightforward computational methods for finding the edit distance between two strings have been used on a variety of problems in biology, genetics, speech and handwriting analysis ([Sankoff and Kruskal, 1983]), as well as in syntactic analysis of formal languages ([Lu and Fu, 1977]). (For a good introduction with applications to many domains, see [Sankoff and Kruskal, 1983].)

Number of Operations.

The edit distance can be defined as the minimum number of token insertions, deletions, and substitutions required to transform one sequence to another [Wagner and Fisher, 1974]. Letting A, B be arbitrary sequences and x, y be tokens, the insertion, deletion, and transformation operators are defined as

$$\begin{array}{ll} \text{insertion:} & AB \xrightarrow{o_i} AxB \\ \text{deletion:} & AxB \xrightarrow{o_d} AB \\ \text{substitution:} & AxB \xrightarrow{o_s} AyB \end{array} \tag{4.8}$$

The *edit distance* is the length n of the shortest sequence of operators $A \xrightarrow{o_1} A_1 \xrightarrow{o_2} \ldots \xrightarrow{o_n} A_n = B$ which transforms token sequence A into B.

For example, consider the following somehwat similar sequences:

$$\begin{array}{lcl} \text{A} & = & \text{[the boy went to the store]} \\ \text{B} & = & \text{[the tall boy drove to the store]} \end{array} \tag{4.9}$$

In this case, A can be transformed to B through the insertion

[the boy] $\xrightarrow{o_i}$ [the *tall* boy]

and the substitution

[went] $\xrightarrow{o_s}$ [drove]

This is a total of 2 operations.

Many different sequences lead to the same transformation. For instance, multiple tokens could have been added to any position in the sequence, and then later removed. One could construct a transformation of an arbitrarily large number of operations in this way. The edit distance is defined as the *minumum* number of operations necessary. The minimum transformation for this example is readily apparent, but in general, finding the minimum sequence of operators is not always this obvious.

Generalizations.

Another transformation sequence, in which the substitution operations "[went] $\xrightarrow{o_s}$ [drove]" is replaced by the two operations of deleting 'went' and inserting 'drove' seems just as basic, but would involve one extra operation, may seem just as basic. The substitution operator can be seen as a generalization over the complex operation of 'delete then insert'.

One generalization of the basic edit distance which solves this discrepancy, as well as increases the power of the measure, is formed by assigning a cost to each operation. One could assign a cost of, say, 1, to each of the insert and delete operators, and a cost of 2 to the substitution operator. Then the two sequences

$$[\text{went}]\xrightarrow{o_d}\xrightarrow{o_i}[\text{drove}][\text{went}]\xrightarrow{o_s}[\text{drove}] \tag{4.10}$$

have an identical cost of 2.

This can be generalized still further, by assigning costs based not only on the operation, but also on the particular symbol involved. This is especially useful in the present linguistic context. For instance, we may not wish to assign a great cost to the insertion of a modifier, such as an adjective, because we know that the structural properties of the NP it modifies will not be unduly affected. For the most part, most contexts that allow a bare N will also allow one or more modifiers. However, we may wish to assign a greater cost to the insertion of a verb, because we know that most constructions do not allow extra verbs to be inserted. Auxilliaries and modals would be assigned a lower insertion cost, because we know they may or may not appear in a VP construction.

Substitutions may be similarly dependent on the words involved. Replacing a word with a like term should not incur a large penalty. However, replacing a word with an unlike word may destroy the linguistic properties of the sequence, and should carry a greater cost. In our example, replacing the general term 'went' with the more specific

'drove' preserves much of the meaning of the sentence. However, replacing 'went' with something completely dissimilar, say 'mantlepiece', would completely destroy the linguistic structure of the sentence, as well as its meaning, and should carry a higher cost.

The *generalized edit distance* can now be defined in terms of the *minumum total cost* of a transformation sequence, rather than simply the minimum number of operations. If one makes the simple assumption that a substitution costs no more than the equivalent deletion-insertion pair, then this minimum cost distance can be shown to obey metric properties (including the triangle inequality). We foramlize this measure in the following section.

The relative costs above are, of course, simply our judgements of how these costs might compare. A more principled method of assigning them is surely needed. An automatic process would be also be beneficial, because of the large number of weights to be assigned. Such a method will be presented in ch. **??**.

Computing edit distance.

There is a straightforward method for computing edit distance ([Sellers, 1974, Wagner and Fisher, 1974]). As a prime example of dynamic programming, one can compute the edit distance for every pair of initial subsequences of the two strings under study, combining results for shorter substrings to give results for longer subsequences.

Explicitly, let our two sequences be $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_n)$, where a_i is the i^{th} token in string A, starting with token 1. We add a *null token* to the head of the each string, tokens a_0 and b_0, representing an empty position that serves as a placeholder. Let us also define the initial substring of a string to be the first i tokens, $A_i \stackrel{\text{def}}{=} (a_0, a_1, \ldots, a_i)$, including the null token inserted at the beginning.

Let $D(A, B)$ by the edit distance between the sequences A and B. The computation starts by defining $D(A_0, B_0)) = 0$, the cost of transforming a_0 to b_0, the null

$A \longrightarrow$

$B \downarrow$

	–	the	boy	went	to	the	store
–	0	1	2	3	4	5	6
the	1	0	1	2	3	4	5
tall	2	1	2	3	4	5	6
boy	3	2	1	2	3	4	5
drove	4	3	2	3	4	5	6
to	5	4	3	4	3	4	5
the	6	5	4	5	4	3	4
store	7	6	5	6	5	4	3

Figure 4.1: Dynamic programming for edit distance ('–' is the null token)

token to itself. Each subsequent step in the computation proceeds with the simple rule:

$$D(A_i, B_j) \stackrel{\text{def}}{=} \min \begin{cases} D(A_i, B_{j-1}) + D_{\text{insert}}(b_j) \\ D(A_{i-1}, B_j) + D_{\text{insert}}(a_i) \\ D(A_{i-1}, B_{j-1}) + D_{\text{substitute}}(a_i, b_j) \end{cases}$$

where $D_{\text{insert}}(x)$ is the cost for inserting x, and $D_{\text{substitute}}(x, y)$ is the cost of substituting x for y. We assume that the substitution is symmetrical, $D_{\text{substitute}}(x, y) = D_{\text{substitute}}(y, x)$

Starting with $D(0, 0)$, one can fill each $D(i, j)$ in a table, ending at $D(m, n)$, the edit distance between the two strings. The table is filled from upper left to lower right, as each entry is computed from its upper, leftward, and diagonal neighbors using the minimum rule above. Figure 4.1 gives this table for the example strings from (4.9).

4.1.4 Linguistic plausibility of comparisons

One can see that all of these measures fulfill a number of our criteria as a measure suitable for measuring some linguistic factors relating two sequences. The Jaccard

and pairwise measures, though, fail in a number of respects to fully capture the kinds of configurational similarity we seek to measure. Jaccard offers the flexibility of using implicit comparisons between *any* pair of tokens from the two sequences, but does this at the expense of ignoring all configurational information. The pairwise measure allows only those comparisons which are aligned in the strictest sense, thus preserving the most obvious configurational information, but, as we saw with example 4.6, it ignores a large class of interesting relations involving mis-alignments.

The edit distance, on the other hand, can allow both structural and token identity disagreements while not being brittle in either. Like the pairwise measure, it will give higher similarity to structurally similar sequences. However, it will degrade gradually, but not break in comparisons which are misaligned. And while it does not go the full way toward incorporating comparisons of *any* pair of tokens as Jaccard does, it uses the comparison between all pairs to decide the best minimal set of transormations. The edit distance certainly does not encode all the linguistic similarity present between two sequences, but it is closer than the others presented here.

4.2 Richer measures

These measures do to capture much of what we mean by linguistic structural similarity, even though they do not incorporate anything that we might regard as a model of syntactic structure. They capture configurational information such as word order, phrasal similarity under modifier insertion, and the similarity of substitution by synonymous terms. We can improve upon them, however, by adding in other important linguistic features. Most importantly, the idea of locality of function, that a word's contribution will be expressed most stringly in the configuration immediately surrounding it, is lacking in the current measures. The following section elaborates some methods for adding this information.

4.2.1 Locality for string measures

There is a degree to which formal configurational (syntactic) considerations are local in nature. As was discussed for the choice of local context as a descriptor for the formal constraints operating on a token in the language, syntactic well-formedness and syntactic interactions are behaviors that operate primarily in a local region. Discussions of long-distance dependencies as deviation from the norm only serve to to highlight the degree to which local interactions are assumed as the norm.

In order to highlight the locality of these configurational dependencies, we can modify the importance asociated with each of the token comparisons contributing to the distance measure. One would like to weight near tokens more heavily, but without ignoring the contributions of distant ones. The window of our short context sequences is already a simple version of such a choice, including near tokens up to some distance, and completely ignoring forther ones.

A linear weight decreasing with distance from t_0 would include contributions from all tokens up to a point. We could, for instance, simply count the nearest token in full, and decrease the weight accorded to token t_i linearly. If the maximum weight is $w = 1$, the weight for each token t_i under this approach is given by $w_i = 1 - iK$, for some constant $0 < K < 1$. One has to be sure that this decrease is not too rapid, however, since the point at which the wieght becomes zero is the effective width of the context description. It would be counterproductive to include more distant tokens, as they would have negative weights.

Alternatively, one could decrease the weight geometrically with each token, such that the weight given to a token is a fixed fraction of the previous weight. In this geometric approach, the weight is given as $w_i = K^i$, again for $0 < K < 1$. This approach has the advantage that the weights are positive for all i.

Half-weight length.

For either the linear or the geometric approach, we can specify the rate of drop off in terms of a *half weight length*, P_h, which is the length (token position) at which the contribution of a token drops to 1/2. This correspondence between length and weight drop off allows us to connect the weighting constants with our observations about the range of interaction included in the measure.

For the linear scheme, the dependence of the weighting factor, K, on this half weight length is:

$$w_{P_h} = 1/2 = 1 - P_h K \quad \Longrightarrow \quad K = 1/2P_h \tag{4.11}$$

and for the geometric shceme:

$$w_{P_h} = 1/2 = K^{P_h} \quad \longrightarrow \quad K = (1/2)^{\frac{1}{P_h}} \tag{4.12}$$

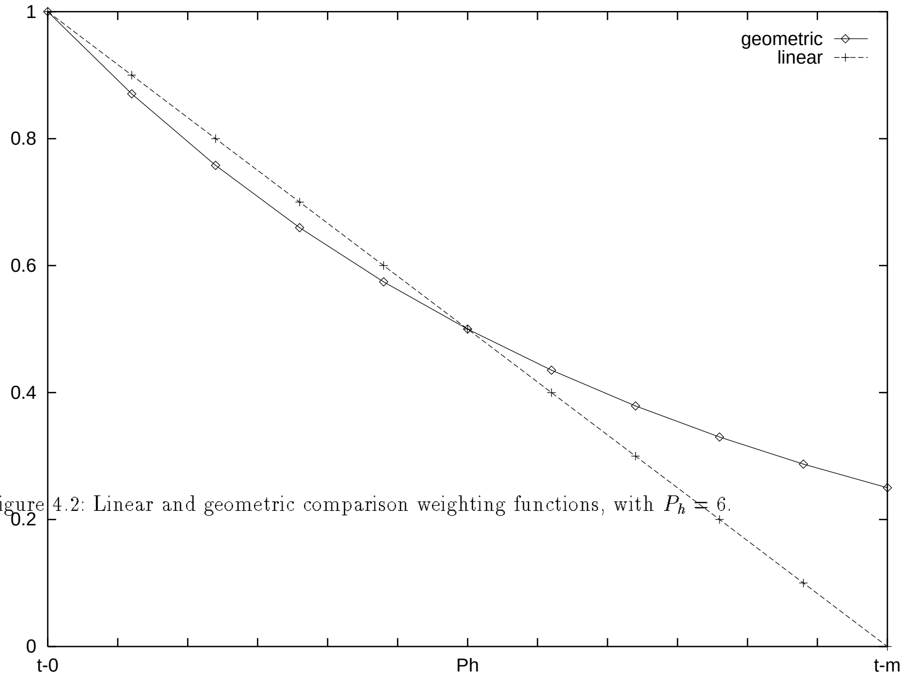


Figure 4.2: Linear and geometric comparison weighting functions, with $P_h = 6$.

The figure (4.2) illustrates the properties of the weighted dropoff for both the linear and geometric approaches, with $P_h = 6$. The geometric scheme has weights approx. 5-10% lower for the first few tokens, but then flattens out after P_h as the linear scheme approaches zero.

[summary]

Locality for the set measure

Locality dropoff weighting can be incorporated into the set measures described above. Recall the simple set measure $S = \|A \cap B\|$. We can weight each element of the intersection by the product of the weights on each occurrence of that element as a token of A and B. Thus if $x \in A \cap B$, and the occurrences of x are $a_2 = x$ and $a_3 = x$, then the weight given to x in the intersection would be the product $w_2 w_3$.

Formally, we can define the weighted set measure S' as:

$$
\begin{aligned}
S' &\stackrel{\text{def}}{=} \sum_{t \,\epsilon\, A \cap B} s'(t) \\
s'(t) &\stackrel{\text{def}}{=} s'_a(t) s'_b(t) \\
s'_a(t) &= \prod_i \begin{cases} w_i \text{ if } a_i = t \\ 1 \text{ otherwise} \end{cases}
\end{aligned}
\tag{4.13}
$$

This formulation reduces to the cardinality of the intersection ($\|A \cap B\|$) when $w_i = 1$ for all i.

Locality for Jaccard

This weighted set measure can be extended to create a weighted Jaccard measure defined in eqn. 4.1. The weighted extension for the union $A \cup B$ is defined by weighting the occurence of each token by the weight associatied with its position in each respective sequence. This can be defined, using the definition of $s'_a(t)$ given above in

eqn. 4.13 by the formula:

$$
\begin{aligned}
J'(A,B) &\stackrel{\text{def}}{=} \frac{I'(A,B)}{U'(A,B)} \\
I'(A,B) &\stackrel{\text{def}}{=} S' = \sum_{t \,\epsilon\, A \cap B} s'(t) \\
U'(A,B) &\stackrel{\text{def}}{=} \sum_{t \,\epsilon\, A \cup B} s'(t)
\end{aligned}
\tag{4.14}
$$

Thus the locality weighted Jaccard measure is normalized by the weighted sum of all elements of both sequences.

Locality for pairwise comparisons

The pairwise measure has perhaps the easiest modification for locality weighting of the measures given here. In this case, each comparison pair is simply scaled by the weight associated with its position. The weighted pairwise comparison measure, P', is then the sum of the weights of each matching pair:

$$
P'(A,B) = \frac{1}{n} \sum_{i=1}^{n} d_i, \quad \text{where } d_i = \begin{cases} 0 & \text{if } a_i = b_i \\ w_i & \text{if } a_i \neq b_i \end{cases}
\tag{4.15}
$$

Locality in the edit distance

The implementation of this weighted dropoff requires only a small change to the original dynamic programming algorithm for edit distance. The table-filling rule now becomes:

$$
D(A_i, B_j) = \begin{cases} D(A_i, B_{j-1}) + w_{i+j} D_{\text{insert}}(b_j) \\ D(A_{i-1}, B_j) + w_{i+j} D_{\text{insert}}(a_i) \\ D(A_{i-1}, B_{j-1}) + w_{i+j} D_{\text{substitute}}(a_i, b_j) \end{cases}
\tag{4.16}
$$

where w_i is the weight for token distance i. Becase the weights must be indexed off the sum of the indices, $i+j$, the weights decrease twice as fast as might be expected. A pair of contexts with a single insertion needed at token k would differ by $w_{2k} D_{\text{insert}}(a_k)$. However, a substitution required at the same location would be given a weight $2w_{2k}$ (if $D_s = 2D_i$, as is usual).

4.2.2 Token weights

These distance measures can also be generalized to compensate for different similarities between tokens. For instance, if one decides that 'went' and 'drove' are more similar to each other than either is, say, to 'boy' or 'to', then it would be reasonable to have the substitution of 'went' → 'drove' have a lower cost than the other possible substitutions.

4.2.3 String alignments

As a by-product of the edit distance computation, one can create an *alignment* of the two strings, which gives the elementwise pairing that is associated with the minimal sequence of edit operations. This alignment matches the elements of the two sequences in linear order and shows the correspondence between tokens and substrings of the two matched strings. Such an alignment can be generated directly from the table created in the edit distance computation by following the path of minima chosen during the computation from the upper left corner to the lower right. Rightward travel along this path corresponds to insertion of a token from string A, downward travel to tokens from string B, and diagonal paths to substitutions. Multiple minimum paths can occasionally result, giving alternate but equivalent alignments. In these cases, I have chosen to favor the shortest path, which gives precedence to diagonal substitution paths.

The alignment created from our two example strings (figure 4.3) gives the correspondence between the tokens of the two initial strings. From the figure, it is easy to see the structural similarities of the two strings.

–	the	–	boy	went	to	the	store
–	the	tall	boy	drove	to	the	store

Figure 4.3: A string alignment table

Alignments can be created for sets of more than two strings. These can be represented as extended alignment tables, with added rows corresponding to the multiple strings. These alignment tables could further be abstracted to probabilistic descriptions of the sequences, using either a first-order or Markov chain description. Chan and Wang ([Chan and Wang, 1991]) have used *syntheses*, zero-order probabilistic descriptions of alignment tables in order to generalize the edit distance and capture the notion of distance between two sets of sequences. Techniques such as this may prove useful in later work.

4.3 String measure experiments

In order to evaluate the potential usefulness of these metrics for structuring the data space, several experiments were conducted using the various string measures. Each of the metrics was used to evaluate a large sample of real language data taken from a collection of articles from the *Wall Street journal* [?]. Section ?? briefly describes the data and its preparation. See appendix ?? for the full details.

The principle evaluation consisted of computing the value each of the measures on all pairs taken from a set of approx. $N = 10^4$ sequences chosen randomly from the WSJ corpus. This results in $N(N-1)/2$ pairs for which the distance was computed using each measure.

In addition to the corpus of real English sequences, a baseline comparison corpus was created using a first-order distributional approximation. The comparison between this baseline and the actual English data reveals how the structural characteristics of actual language data differ from randomly constructued sequences. The mannar in which the various metrics reveal this structure is also demonstrated.

4.3.1 English sequence data

A short description of the data and it treatment is given here. For a full description of the data, see appendix ??.

The sequences used in the experiments presented in this chapter were drawn form a corpus of *Wall Street Journal* articles, originally published in 1989, and distributed in both an unfiltered and a parsed, part-of-speech tagged forms by the Linguistic Data Consortium [?].

Tokenization

The text has been tokenized to distinguish words from whitespace, punctuation, etc. This processing is somewhat different from that commonly seen in corpus processing literature in that whitespace, punctuation, and other symbols *have been retained as tokens*. It was considered that, as a study of the formal properties of written language, these characteristic demarcations, which are unique to written language, and which form a substantial part of its structure, should be explicitly retained and analyzed along with the words which have explicit counterparts in spoken language. Some properties of the written form, such as the upper/lower case distinction, were not retained in the tokenization process.

The following sequences of characters are tokenized as discrete symbols:

1. any continuous sequence of the alphabetic characters [a-zA-Z]
2. any continuous sequence of whitespace characters [newline space]
3. individual numerals [0-9]
4. individual punctuation characters and other symbols [, . " $ % etc.]

Table 4.1 gives some examples of how the English data and the tokenized sequences correspond.

Defining sequences

Chapter 3 defined the local (n, m)-context of a word in corpus as the sequence:

$$[t_{i-n} - t_{i-1}] \quad t_i \quad [t_{i+1} - t_{i+m}] \tag{4.17}$$

The sequences compared in the experiments presented here are $(7, 7)$-contexts taken from the tokenized WSJ corpus. Each sequence is further broken down into its left and right segments, and comparisons are performed separately on each segment, and then combined. Thus, the $(7, 7)$-context A at position i in the corpus is given as:

$$A = A_L([t_{i-7} - t_{i-1}]) \quad t_i \quad A_R([t_{i+1} - t_{i+7}]) \tag{4.18}$$

Each sequence distance was computed separately on the left and right subsequences of the context. The final measure between two contexts A and B is taken as the sum of these separate left and right measures. The measure $M(A, B)$ between two contexts is computed as

$$M(A, B) = M(A_L, B_L) + M(A_R, B_R)$$

4.3.2 Generated sequence data

A comparison set was created, in order to study the difference between the sampled English data discussed above, and a generated sample with the same first-order distributional properties. The behavior of the various measures with respect to the real

<table>
<tr><th>English</th><th>Tokens</th></tr>
<tr><td>Sales of medium-sized cars, which
Chrysler Corp.'s Chrysler division.
mirrors the "closed-end
dropped nearly 7% Oct. 13,</td><td><code>sales|␣|of|␣|medium|-|sized|␣|cars|,|␣|which</code>
<code>chrysler|␣|corp|.|'|s|␣|chrysler|␣|division|.</code>
<code>mirrors|␣|the|␣|"|closed|-|end</code>
<code>dropped|␣|nearly|␣|7|\%|␣|oct|.|1|3|,</code></td></tr>
</table>

Table 4.1: English data v. tokenized form. Tokens are separated by '|', while '␣' indicates an explicit whitespace token.

and generated data should give information about the nature of the relations that underly the comparison measures.

The sample was created using a token generator which outputs tokens according to an ideal Zipf distribution, $p(\mathrm{w}) = K1/r(\mathrm{w})$. The constant K was chosen so that the total number of tokens The log-log plot of token frequency versus token rank is shown in figure

[I need to figure out whether I need the generated set which has approx the same number of tokens as the real sample, or a se which is generated with parameters for the same *total* number of tokens as the entire WSJ text. Or maybe, I need just to figure out what the fucking slope parameters are, and how to relate those to the generator model.]

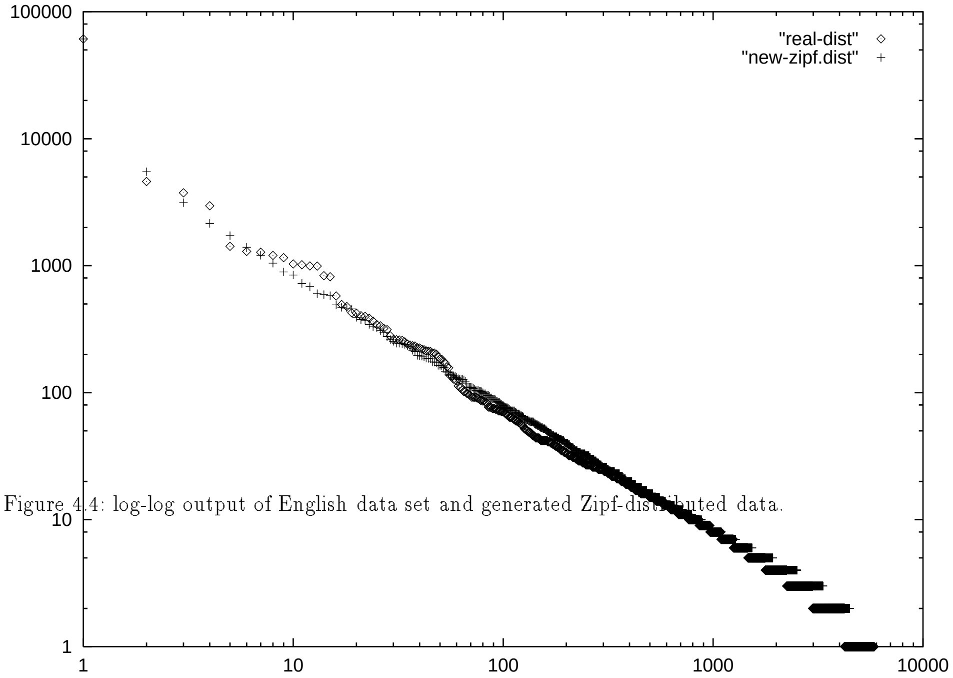


Figure 4.4: log-log output of English data set and generated Zipf-distributed data.

4.3.3 Distance measure results

Histograms of the Jaccard, pairwise, and edit distance measures are shown in figures 4.5, 4.6, and 4.7. Each point on the plots expresses the number of sequence pairs having a given distance measured between them. The total number of pairs in each plot is 42665703 (this is $N(N-1)/2$ for $N = 9238$). As mentioned above, the distance between a pair of sequences is the sum of the left and right side measures taken independently, $M(A,B) = M(A_L, B_L) + M(A_R, B_R)$.

Both the pairwise measure (4.6) and Jaccard measure (4.5) are normalized, with values in the range [0-1], and so the combined left+right measures were binned in the range [0–2], with a bin size of 0.01. Edit distance (4.7) was binned in the range [0–30], also with a bin width of 0.01.

The figures on the ordinate in all the plots are given in the number of bins, and thus are either 10× or 100× the measured distance. The response variable (number of pairs) are plotted on a log scale, so that the enormous number of large distance pairs does not overwhelm the display of small distance results.

Two properties are distinctly evident in all of these plots. Perhaps the most obvious property is the 'spikiness' of the edit distance and pairwise comparison histograms, indicating that these measures have a sparse set of possible values. These measures are quite discrete, and cannot represent the kinds of fine distinctions that a linguistically realistic measure should have. Certainly, in order to be useful, any measure of linguistic similarity will have to have more than a handful of possible values.

The Jaccard measure suffers from this problem to a lesser degree. however, as discussed above, the Jaccard measure suffers from its insensitivity to positional information of any sort.

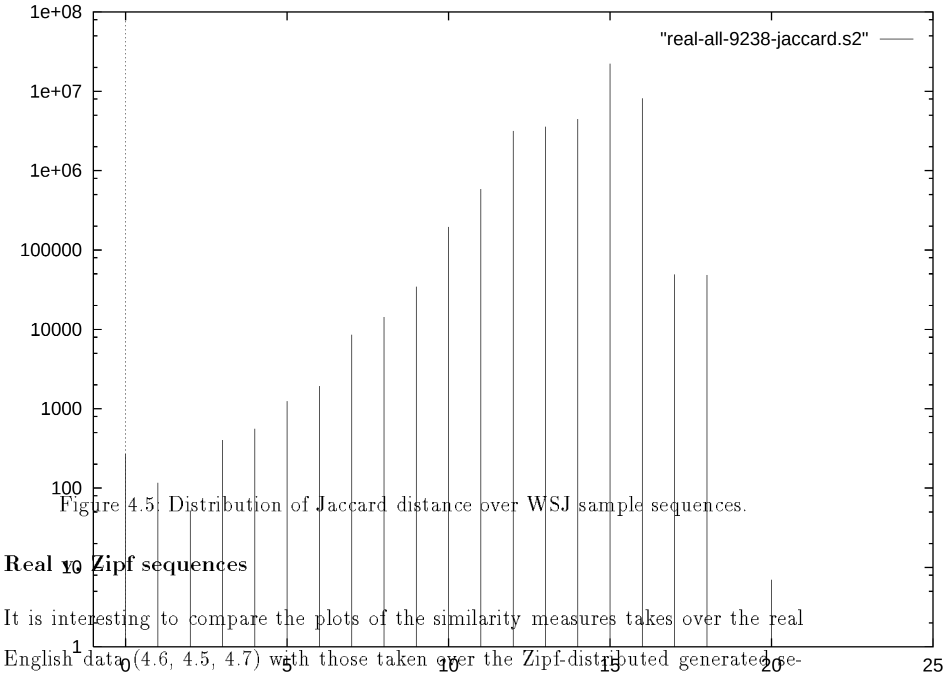


Figure 4.5: Distribution of Jaccard distance over WSJ sample sequences.

Real vs Zipf sequences

It is interesting to compare the plots of the similarity measures takes over the real English data (4.6, 4.5, 4.7) with those taken over the Zipf-distributed generated sequences. Plots 4.8, 4.9, and 4.10 show measures for the real data along with the generated data. In all three plots, it is evident that the real English data contains many more pairs with low distance than the generated data does.

This observation adds evidence to the assertion that the occurrences of words in real human languages are structured and interdependent. Therefore, these interactions and cannot and should not be represented by first order, independent models.

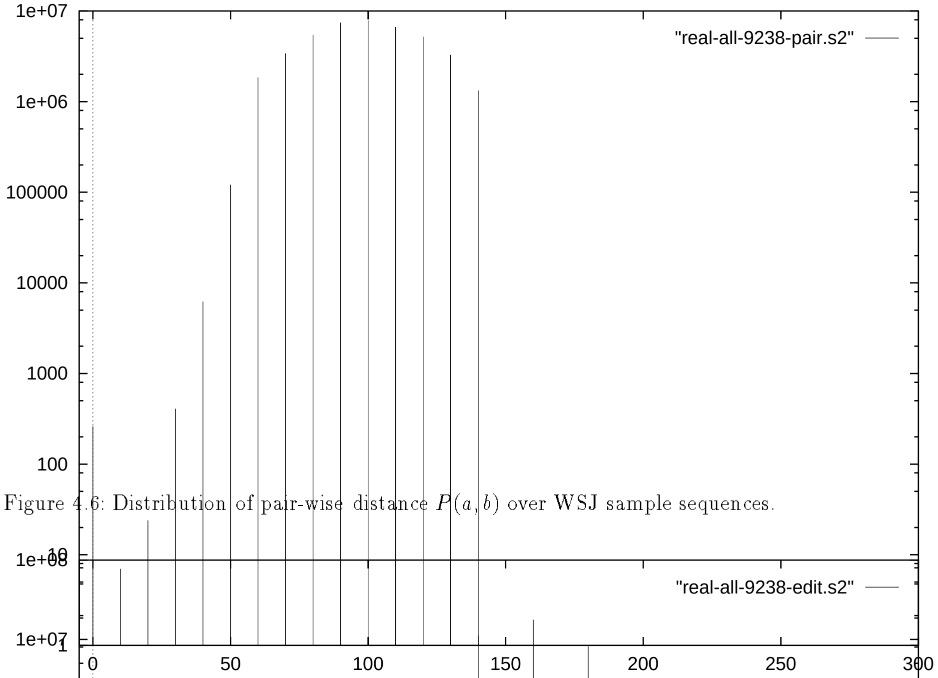


Figure 4.6: Distribution of pair-wise distance $P(a, b)$ over WSJ sample sequences.

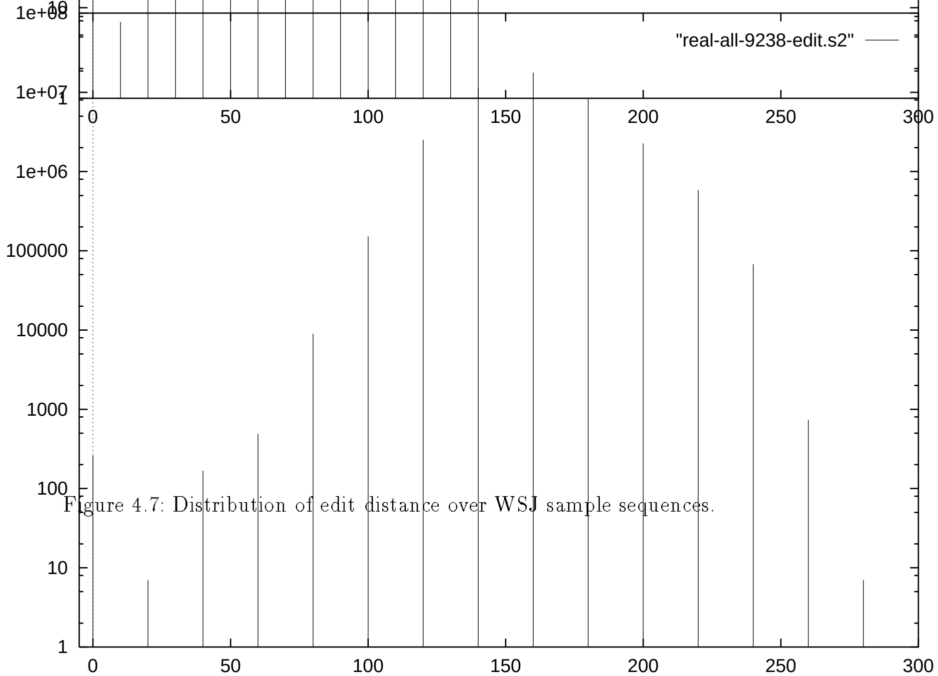


Figure 4.7: Distribution of edit distance over WSJ sample sequences.

Figure 4.8: Jaccard measure for real English sequences and Zipf-distributed generated sequences.

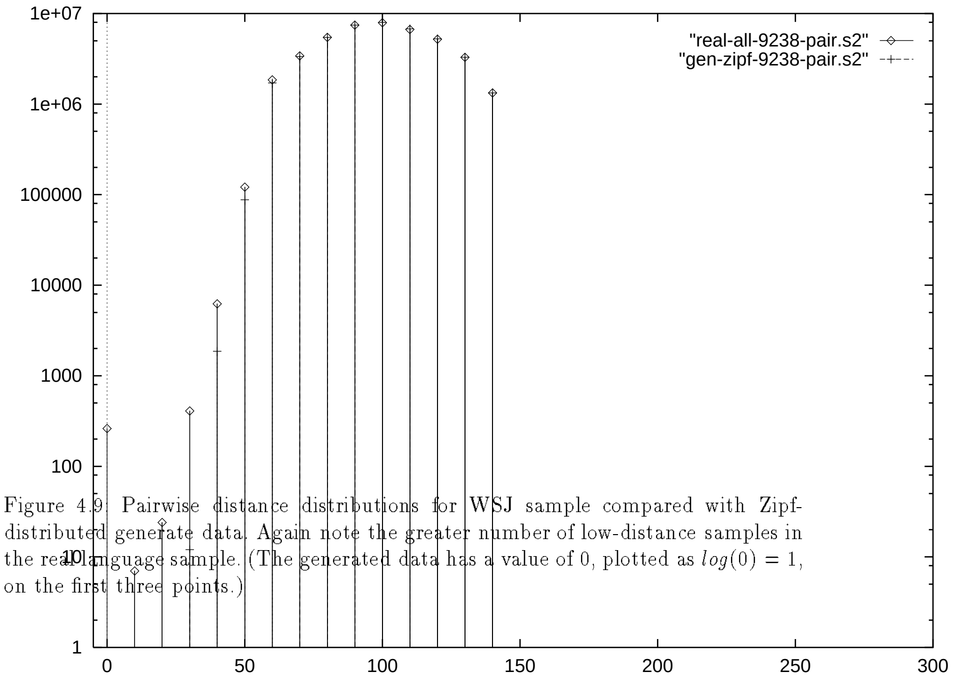


Figure 4.9: Pairwise distance distributions for WSJ sample compared with Zipf-distributed generate data. Again note the greater number of low-distance samples in the real language sample. (The generated data has a value of 0, plotted as $log(0) = 1$, on the first three points.)

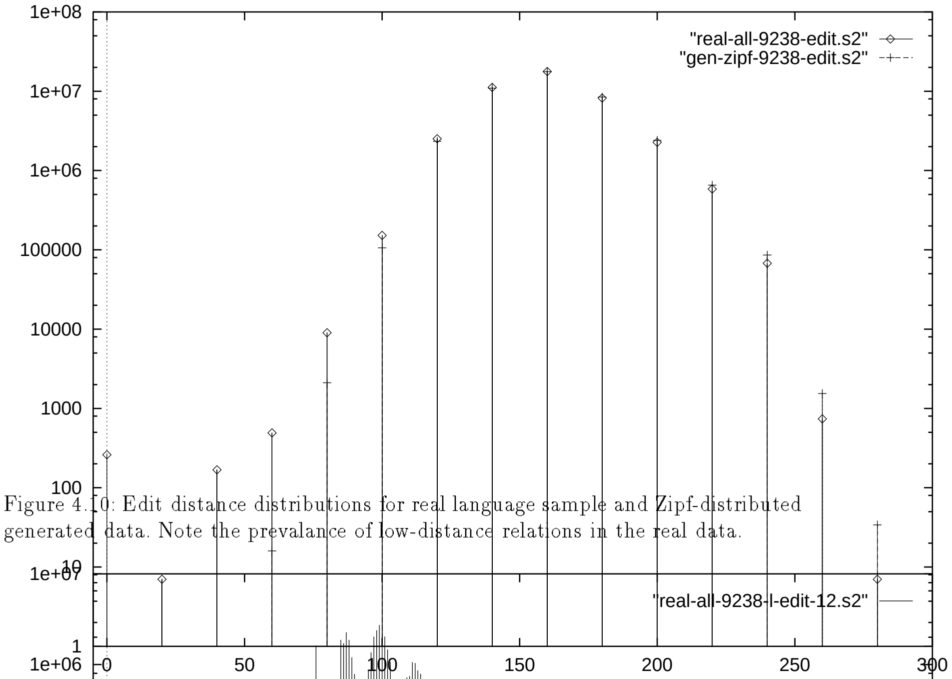


Figure 4.10: Edit distance distributions for real language sample and Zipf-distributed generated data. Note the prevalance of low-distance relations in the real data.

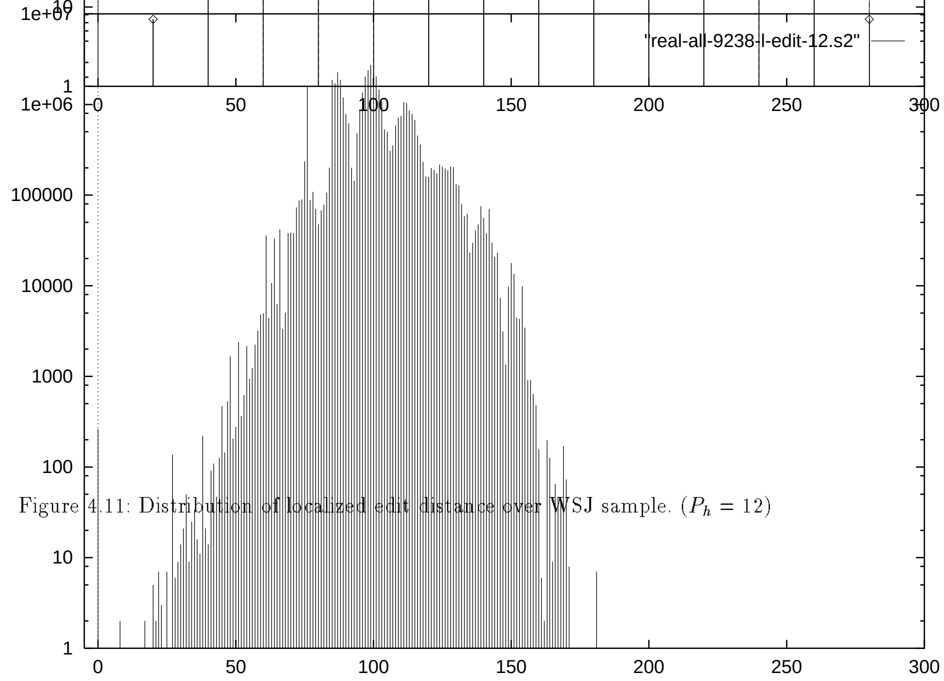


Figure 4.11: Distribution of localized edit distance over WSJ sample. ($P_h = 12$)

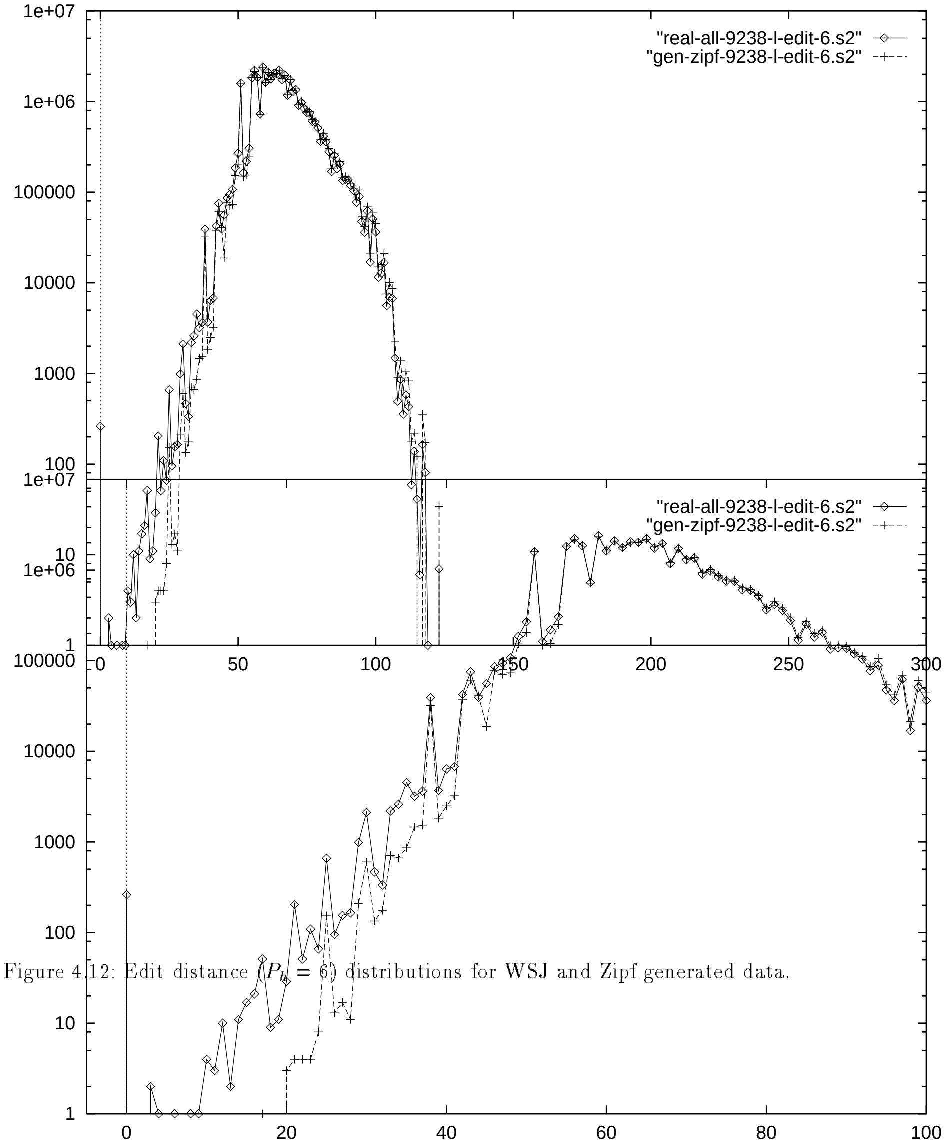


Figure 4.12: Edit distance $(P_h = 6)$ distributions for WSJ and Zipf generated data.

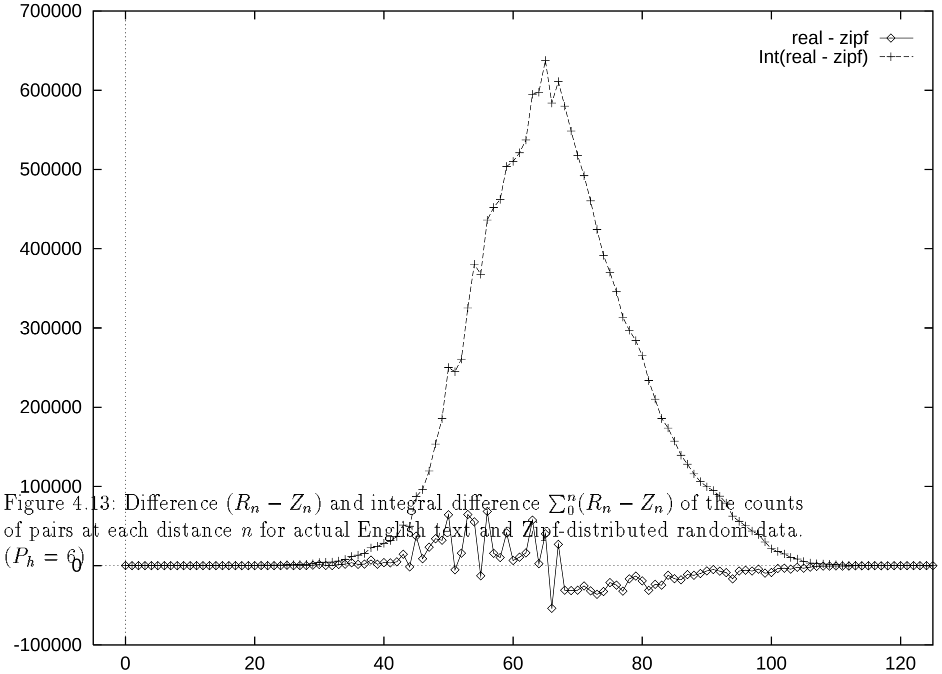


Figure 4.13: Difference $(R_n - Z_n)$ and integral difference $\sum_0^n (R_n - Z_n)$ of the counts of pairs at each distance n for actual English text and Zipf-distributed random data. $(P_h = 6)$

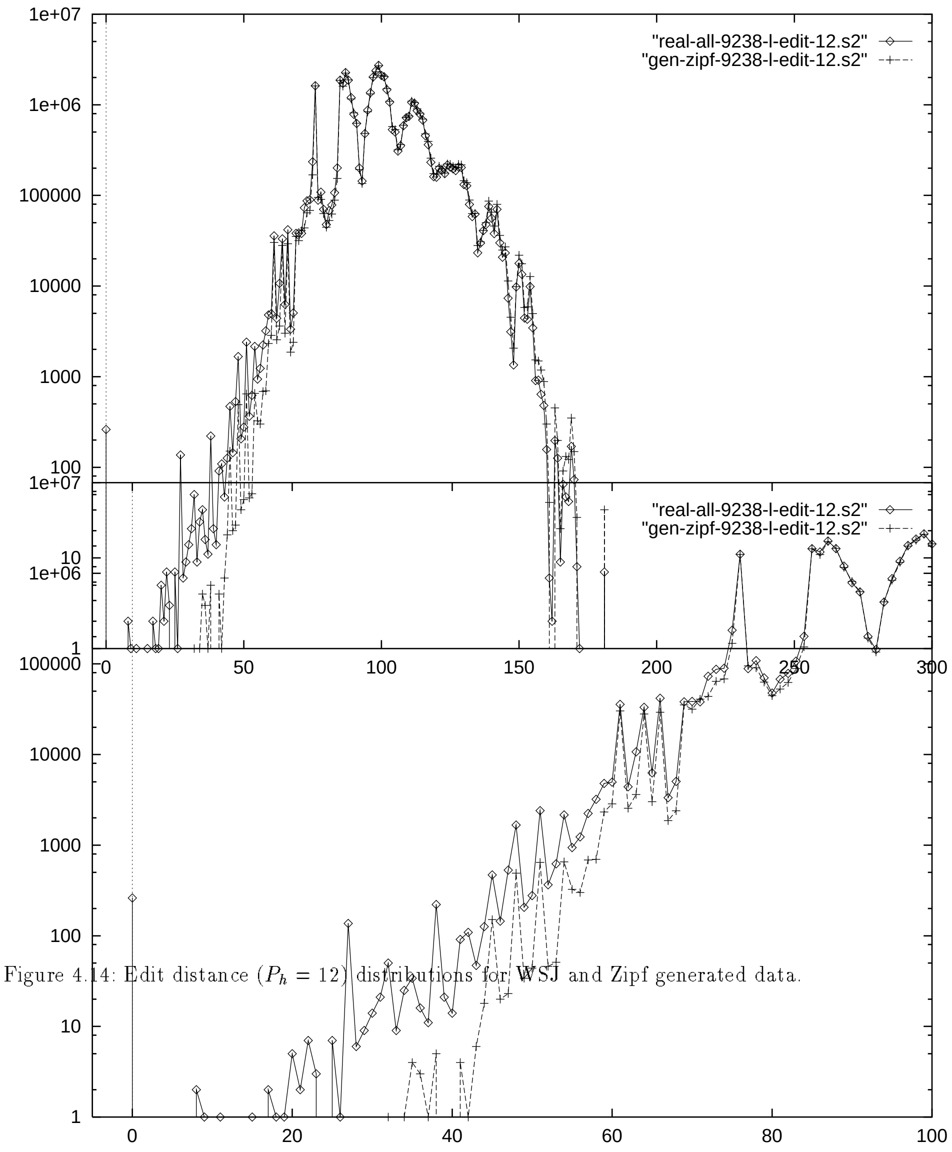


Figure 4.14: Edit distance ($P_h = 12$) distributions for WSJ and Zipf generated data.

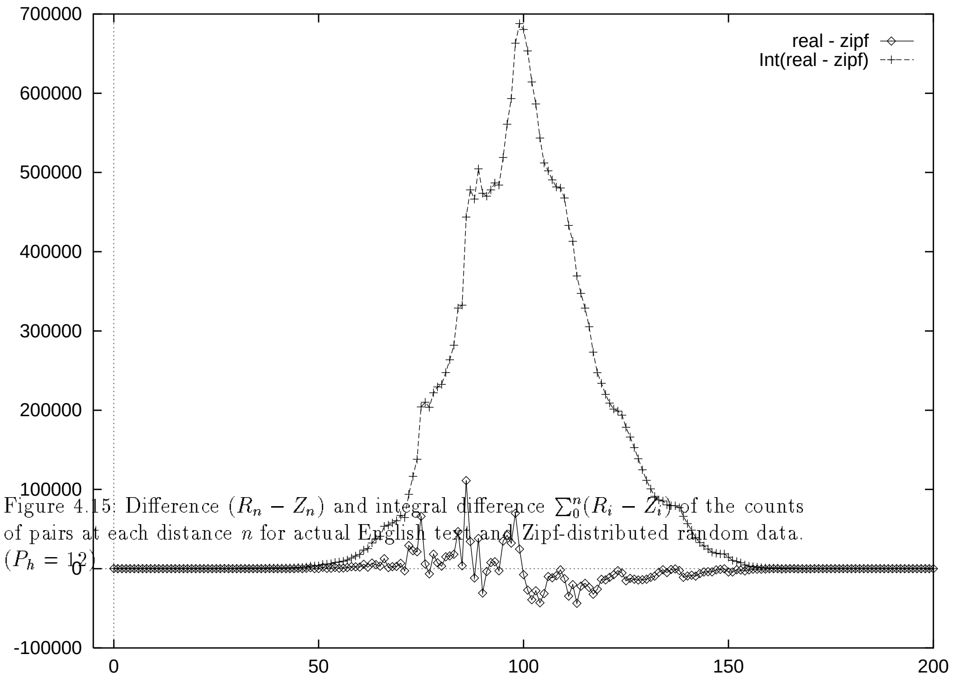


Figure 4.15: Difference $(R_n - Z_n)$ and integral difference $\sum_0^n (R_i - Z_i)$ of the counts of pairs at each distance n for actual English text and Zipf-distributed random data. $(P_h = 12)$.

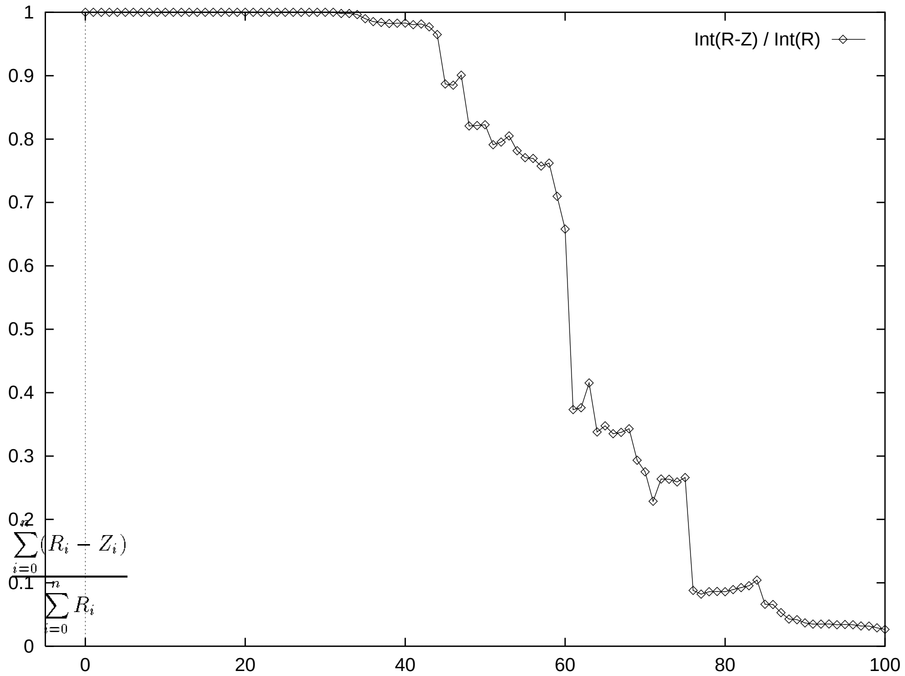


Figure 4.16: Fraction of pairs not accounted for by the random data as a function of distance. (localized edit distance $P_h = 12$)

Chapter 5

Classification and Generalization

[why are we doing clustering?] Why generalize? reiterate motivation, purpose. i.e. – the need to abstract to an understandable level, which will explicate the understanding of the structure underlying the data.

The previous chapters have discussed methods of characterizing sets of contexts, and measures for comparing them, [1] but the question of how one finds the sets has been shrugged off until now. This chapter discusses methods for grouping the raw lexical data into classes which have distinct and interesting properties.

5.1 Characterizing multiple contexts

In order to make comparisons between all the uses of two words, as with the word-based models, or to make comparisons between contexts grouped by other principles, a representation is needed that can encompas multiple contexts.

If we want to consider all the uses of a particular word, k for instance, we can treat them as a set and simply list them. Here is such a list using α's for the left

[1] not entirely true! much of that discussion moved to this chapter!

items, and β's for the right, each context identified by a superscript:

$$\begin{aligned} \alpha_n^1 \ldots \alpha_2^1\, \alpha_1^1\ \mathrm{k}^1\ \beta_1^1\, \beta_2 1 \ldots \beta_m^1 &= c_k^1 \\ \alpha_n^2 \ldots \alpha_2^2\, \alpha_1^2\ \mathrm{k}^2\ \beta_1^2\, \beta_2 2 \ldots \beta_m^2 &= c_k^2 \\ \vdots \\ \alpha_n^l \ldots \alpha_2^l\, \alpha_1^l\ \mathrm{k}^l\ \beta_1^l\, \beta_2 l \ldots \beta_m^l &= c_k^l \end{aligned} \tag{5.1}$$

While this sort of representation contains all the available information, it is not well suited to the tasks of generalization and comparisons; the tabular list makes no abstractions over the original data. In order to make a comparison to a class of contexts represented this way, we would need to compare against every context listed, and average the results by some means. Analysing the data by any means would involve looking at each element separately, and and fails to capture the abstraction desired in forming such a categry in the first place.

5.1.1 Simply conditioned models

There are many ways to reduce the size of this set description, and make it more managable. One family of such reduction I will call the *simply conditioned models*.

The first such model, the 0^{th}-order conditioned model, is the easiest to describe. For a set of (n, m)-contexts, each of the $n + m$ positions is represented by a vector of the probabilities of an item occurring in that position:

$$\ldots \begin{bmatrix} p(\alpha_2 = w_1) \\ p(\alpha_2 = w_2) \\ p(\alpha_2 = w_3) \\ \vdots \end{bmatrix} \begin{bmatrix} p(\alpha_1 = w_1) \\ p(\alpha_1 = w_2) \\ p(\alpha_1 = w_3) \\ \vdots \end{bmatrix} \; k \; \begin{bmatrix} p(\beta_1 = w_1) \\ p(\beta_1 = w_2) \\ p(\beta_1 = w_3) \\ \vdots \end{bmatrix} \begin{bmatrix} p(\beta_2 = w_1) \\ p(\beta_2 = w_2) \\ p(\beta_2 = w_3) \\ \vdots \end{bmatrix} \ldots \tag{5.2}$$

If we replace each of the vectors of positional probability with a simpler notation, we can write this more concisely as

$$\cdots\ Pr(\alpha_2)\ \ Pr(\alpha_1)\ \ k\ \ Pr(\beta_1)\ \ Pr(\beta_2)\ \cdots \tag{5.3}$$

Of course, this representation suffers a few problems. One is that for natural languages the number of possible words, the number of possible α's and β's, is potentially infinite; this makes the representation for a finite set of contexts an infinite array of probabilities. Even with a fixed vocabulary of size V, we need $V(n+m)$ parameters for each set of contexts. This can be remedied by including only probabilities for those items that have been seen, leaving any excluded values implicitly zero. Another approach is to only compute probabilities for some select set of items that the researcher feels has indicative power. For instance, [Schütze, 1993] uses onely the 5000 most frequent words of English in his study, while [Hughs, 1994] uses only the 25 most frequent items. Other possibilities include high frequency function words, such as prepositions, case markers, and determiners.

A more serious problem is the loss of useful information in this representation. None of the probabilities given for the positional values are at all correlated in the representation. This means, for instance, that the two sets of $(0,2)$-contexts

$$s_1 = \{kAB, kBA\} \qquad s_2 = \{kAA, kBB\} \tag{5.4}$$

have equivalent representations

$$\begin{bmatrix} p(A) = .5 \\ p(B) = .5 \end{bmatrix} \quad k \quad \begin{bmatrix} p(A) = .5 \\ p(B) = .5 \end{bmatrix} \tag{5.5}$$

Even with this deleterious loss of correlation, this sort of representation has worked fairly well for a few word-based classifications ([Schütze, 1993],[Hughs, 1994],[Finch and Chater, 1992], [Futrelle and Gauch, 1993]).

[Maybe just change this to a ref. to a related work section]

All of these include studies of $(2,2)$-contexts represented as 0^{th}-order conditioned models. In each study, one set of contexts was used for all the occurrences of a given word. The conditioned models for every set were compared, and classes of contexts were formed which corresponded to categories of words, all having similar contexts.

These studies used various similarity measures, such as manhattan metric ([Hughs, 1994], vector cosine measures ([Hughs, 1994], [Futrelle and Gauch, 1993]), cosine measures over a reduced space ([Schütze, 1993]), or entropy measures ([Chan and Wang, 1991]. Various clustering methods were used to form categories based on these measures.

Higher order conditioning

The information loss of this simple vector model can be reduced through conditioning the probabilities. Since I have argued that the most local interactions are the most important, it makes sense to condition the let and right sequences on the positions closest to the key word in the center.

In general, we can call a sequence conditioned on the first u positions to the left and v positions to the right a (u,v)-*simple conditioned model.* Of course, positions closer to center than u or v would be conditioned only on those positions closer to center.

Thus in representing a postion, β_n, we would need to give its probabilities conditioned on the $j_{n,v} = \min(v, n-1)$ previous positions:

$$Pr(\beta_n|\beta_1\beta_2\ldots\beta_j = w_1w_2\ldots w_j), \; j = \min(v, n-1) \tag{5.6}$$

(Note that this is not the standard notation used for in n-gram analyses or Markov modelling. In this notation, $Pr(\beta_2|\beta_1 = w_1)$ is the distribution of the second position, β_2, only for those cases in which $\beta_1 = w_1$. Similarly, $Pr(\beta_3|\beta_1 = w_1)$ is the distribution of the third position, β_3, when the *first* position is $\beta_1 = w_1$.)

This representation requires $V^{j_{n,v}+1}$ parameters for each position: a probability for each of V terms, each conditioned against $V^{j_{n,v}}$ possible sequences of preceding terms.

It is obvious that a set of $(u+1, v+1)$-contexts is required to build a (u,v) model.

As an example, the $(0,1)$ and $(0,2)$-simply conditioned models on $(0,3)$-contexts would look as follows:

$$k \begin{bmatrix} p(\beta_1=w_1) & [Pr(\beta_2|\beta_1=w_1)\ Pr(\beta_3|\beta_1=w_1)] \\ p(\beta_1=w_2) & [Pr(\beta_2|\beta_1=w_2)\ Pr(\beta_3|\beta_1=w_2)] \\ \vdots & \end{bmatrix} \tag{5.7}$$

$$k \begin{bmatrix} p(\beta_1=w_1) & \begin{bmatrix} p(\beta_2=w_1|\beta_1=w_1) & Pr(\beta_3|\beta_1\beta_2) \\ p(\beta_2=w_2|\beta_1=w_1) & Pr(\beta_3|\beta_1\beta_2) \\ p(\beta_2=w_3|\beta_1=w_1) & Pr(\beta_3|\beta_1\beta_2) \\ \vdots & \end{bmatrix} \\ p(\beta_1=w_2) & \begin{bmatrix} p(\beta_2=w_1|\beta_1=w_2) & Pr(\beta_3|\beta_1\beta_2) \\ p(\beta_2=w_2|\beta_1=w_2) & Pr(\beta_3|\beta_1\beta_2) \\ p(\beta_2=w_3|\beta_1=w_2) & Pr(\beta_3|\beta_1\beta_2) \\ \vdots & \end{bmatrix} \\ \vdots & \end{bmatrix} \tag{5.8}$$

5.1.2 Lattice conditioned models

5.2 Modelling the edit operations

Explicit models of multiple context can also be achieved by explicitly modelling the edit operations that are used to compare the group memebers.

Refer to Haussler et al. paper which

5.3 Background in Clustering Methods

Clustering techniques are methods for grouping data into distinct categories. Of course, the notion of *distinct* is dependent on what you care to pay attention to, and perhaps what you are able to pay attention to. In general, clustering techniques

fall into several families, among them parametric, hierarchical, and density-driven methods.

5.3.1 Parametric

Parametric clustering can be seen as an optimization process, in which clusters are found which maximize some fitness criterion over the data. Often, this criterion takes the form of a combination of within and between group similarity.

algorithms operate by positing a model for form of the groups or clusters to be found,

5.3.2 fixed

Fixed models (k-means) as a middle ground

5.3.3 Hieracrchical

divisive

Top-down methods, neccessity of descriminants, computational complexity.

agglomerative

Bottom-up methods. Requires metric, or at least a partial ordering function. computational complexity.

5.4 Context Clustering

By choosing the appropriate similarity metrics and models of context, we can use well-known clustering and/or numerical taxonomy techniques to investigate the structure of the language.

5.5 Classification stability and convergence

a discusssion of how an iterated classification behaves.

take a sample chunk of text, say 10-20k words of wsj, with POS tags.

find full distance measure between elements of the text, $D_{ij} = d(t_i, t_j)$ using word identity ($d(a,b) \stackrel{\text{def}}{=} [a \stackrel{?}{=} b]$) as a measure of interword distance

cluster into some pre-set number of classes.

Iteration step:

find full distance set between elements of the text, $D_{ij} = d'(t_i, t_j)$ using *class* identity as a measure of interword distance. $d'(a,b) \stackrel{\text{def}}{=} [\omega(a) \stackrel{?}{=} \omega(b)]$

cluster into smae pre-set number of classes.

measure similarity of classification to previous classification

Repeat classification based on class membership.

1) How does classification change over time?

2) Does classification ever completely stabilize?

Chapter 6

Operations on the model

descibing how to interpret the things (using data visualization) how to compare new individual points to existing models (intercalation – look it up!) how to evaluate models w.r.t other classification schemes. how to add user-vetted interpretation knowledge.

The previous chapter on classification techniques (chap. ??) dealt with the problems of having large numbers of sample points by abstracting from the individual points to classes. This chapter deals with a similar problem, that of having a large amount of data describing those points or classes.

6.1 Reduction of Similarity data

6.1.1 Correspondence anal. methods

PCA

Column (variable) scaling (to equalize variances)

Inverse-scale variables

(introducing extra vars which have inverse scales from the given data.) what does this really do?

6.2 Coordinate transformations

6.2.1 MDS Methods

A set of methods often called multi-dimensional scaling methods are also available for tackling problems of this kind.

6.2.2 Torgerson Metric Scaling

$$T(D) = XX^T \qquad \text{where } \|x_i - x_j\| \approx d_{ij} \tag{6.1}$$

we define the matrix A

$$A \stackrel{\text{def}}{=} T(D) \stackrel{\text{def}}{=} (I - 1/N)D^{(2)}(I - 1/N) \tag{6.2}$$

where $D^{(2)}$ is derived from D by squaring each element.

Equivalently, we can define A using an element-wise definition:

$$A \stackrel{\text{def}}{=} a_{ij} = d_{ij}^2 - d_{\cdot j}^2 - d_{i\cdot}^2 + d_{\cdot\cdot} \tag{6.3}$$

where

$$d_{\cdot j}^2 = \frac{1}{N}\sum_k d_{kj}^2 \qquad d_{i\cdot}^2 = \frac{1}{N}\sum_k d_{ik}^2 \qquad d_{\cdot cdot} = \frac{1}{N^2}\sum_k \sum_l d_{kl}^2 \tag{6.4}$$

Rank Order SSA

6.3 Abstracting Context Data

6.3.1 Representing neighbors as classes

6.3.2 represetnting multi-modal words

Metrics for multi-modal words.

Chapter 7

Evaluating various models

7.1 Predictive accuracy: perplexity

7.2 Comparison to Pasrt-of-Speech tagging

7.3 Word sense identification

stuff

Bibliography

[Anderberg, 1973] Michael R. Anderberg. *Cluster Analysis for Applications.* Academic Press, Inc., London, 1973. originally presented as Anderberg's thesis, University of Texas, Austin, 1971.

[Ass, 1991] Association for Computational Lingistics. *Proceedings of the 29th Annual Meeting of the ACL*, 1991.

[Ayuso *et al.*, 1992] Damaris Ayuso, Sean Boisen, Heidi Fox, Robert Ingria, and Ralph Weischedel. Bbn: Description of the plum system as used for muc-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 1992.

[Baker, 1975] Janet K. Baker. Stochastic modeling for automatic speech understanding. In D. R. Reddy, editor, *Speech Recognition.* Academic Press, 1975.

[Basili *et al.*, 1993] Roberto Basili, Maria Teresa Pazienza, and Paola Velardi. Hierarchical clustering of verbs. In *Acquisition of Lexical Knowledge from Text, Proceedings of the SIGLEX Workshop*, 1993.

[Black *et al.*, 1992] Ezra Black, Fred Jelinek, John Lafferty, David Magerman, Robert Mercer, and Salim Roukos. Towards history-bases grammars: Using richer models of probabilistic parsing. In *Proceedings of the Fifth DARPA Workshop on Speech & Natural Language*, 1992.

[Bloomfield, 1933] Leonard Bloomfield. *Language.* Holt, New York, 1933.

[Boguraev and Briscoe, 1989] Branimir Boguraev and Ted Briscoe, editors. *Computational Lexicography for Natural Language Processing.* Longman, London, 1989.

[Breiman *et al.*, 1984] Leo Breiman, jerome H. Freidman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees.* Wadsworth & Brooks, Monterey, California, 1984.

[Brent, 1991] Michael R. Brent. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the ACL*, 1991.

[Brown *et al.*, 1992a] Peter Brown, Vincent J. Della Pietra, Peter V. DeSouza, Jennifer C. Lai, and Robert L. Mercer. Class-bassed n-gram models of natural language. *Computational Linguistics*, 18(4), 1992.

[Brown *et al.*, 1992b] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 1992.

[Chan and Wang, 1991] S.C. Chan and A.K.C. Wang. Synthesis and recognition of sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(12):1245–1255, 1991.

[Chomsky, 1957] Noam Chomsky. *Syntactic Structures.* Mouton & Co., The Hague, 1957.

[Chomsky, 1965] Noam Chomsky. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, 1965.

[col, 1988] Proc. of the 12th International Conference on Computational Linguistics, Budapest, 1988.

[col, 1992] *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, 1992.

[Cowie *et al.*, 1993] Jim Cowie, Louise Guthrie, James Pustejovsky, Takahiro Wakao, Jin Wang, and Scott Waterman. The diderot information extraction system. In *Proceedings of the First Conference of the Pacific Association for Computational Linguistics*, Vancouver, 1993.

[Dagan *et al.*, 1993] Ido Dagan, Shaul Marcus, and Shal Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting of the ACL*, 1993.

[Dagan *et al.*, 1994] Idao Dagan, Fernando Periera, and Lillian Lee. Similarity-based estimation of word coocurrence probablities. In *Proceedings of the 32nd Annual Meeting of the ACL*, 1994.

[Davison, 1983] Mark L. Davison. *Multidimensional Scaling*. Wiley Series in probability and mathematical statistics. John Wiley & Sons, 1983.

[DeMarcken, 1990] Carl DeMarcken. Parsing the lob corpus. In *Proceedings of the 28th Annual Meeting of the ACL*, 1990.

[Dunn and Everitt, 1982] G. Dunn and Brian S. Everitt. *An introduction to numerical taxonomy*. Cambridge University Press, 1982.

[Edwards, 1972] A. W. F. Edwards. *Likelihood*. Cambridge University Press, 1972.

[Elliot *et al.*, 1995] Robert J. Elliot, Lakhdar Aggoun, and John B. Moore. *Hidden Markov Models*. Number 29 in Applications of Mathematics. Springer-Verlag, 1995.

[Everitt, 1980] Brian Everitt. *Cluster Analysis*. Halsted Press, second edition, 1980.

[Fillmore, 1968] Charles Fillmore. The case for case. In Emmon Bach and R. T. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart & Winston, New York, 1968.

[Finch and Chater, 1992] Steven Finch and Nick Chater. Bootstrapping syntactic categories using statistical methods. In Walter Daelemans and David Powers, editors, *Background and Experiments in Machine Learning of Natural Langague*, 1992.

[Fu and Booth, 1975] King-Sun Fu and Taylor L. Booth. Grammatical inference: Introduction and survey. *IEEE Transactions on Sytems, Man, and Cybernetics*, 5(1,4), 1975.

[Fukunaga, 1990] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., San Diego, second edition, 1990.

[Futrelle and Gauch, 1993] Robert Futrelle and Susan Gauch. Experiments in syntactic and semantic classifications and disambiguations using bootstrapping. In *Acquisition of Lexical Knowledge from Text, Proceedings of the SIGLEX Workshop*, 1993.

[Gale and Church, 1990] William A. Gale and Kenneth W. Church. Poor estimates of context are worse than none. In *Proceedings of the 28th Annual Meeting of the ACL*, 1990.

[Gale, 1986] William A. Gale, editor. *Artificial intelligence and statistics*. Addison-Wesley, 1986.

[Gonzales, 1978] Rafael C. Gonzales. *Syntactic pattern recognition*. Addison-Wesley, 1978.

[Good, 1953] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40, 1953.

[Goodman *et al.*, 1991] Marc Goodman, Scott Waterman, and Rick Alterman. Interactive reasoning about spatial components. In *Proceedings 13th Conference of the Cognitive Science Society*, Hillsdale, NJ, 1991. Earlbaum.

[Greenacre, 1984] Michael J. Greenacre. *Theory and Applications of Correspondence Analysis.* Academic Press, 1984.

[Grishman and Sterling, 1992] Ralph Grishman and J. Sterling. Acquisition of selectional patterns. In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.

[Grishman *et al.*, 1992] Ralph Grishman, C. Macleod, and J. Sterling. New york university: Description of the proteus system as used for muc-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 1992.

[Hindle and Rooth, 1991] Don Hindle and Mats Rooth. Structual ambigity and lexical relations. In *Proceedings of the 29th Annual Meeting of the ACL*, 1991.

[Hobbs and Appelt, 1992] Jerry Hobbs and Doug Appelt. Sri international: Description of the fastus system used for muc-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 1992.

[Hobbs *et al.*, 1990] Jerry Hobbs, Doug Appelt, and M. Pal. Interpretation as abduction. Technical Report Technical Note 449, SRI International AI Center, Palo Alto, 1990.

[Hogeweg and Hesper, 1984] P. Hogeweg and B. Hesper. The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *J. Molecular Evolution*, 20:175–186, 1984.

[Hughs, 1994] John Hughs. *Automatically acquiring a classification of words.* PhD thesis, University of Leeds, 1994.

[Jeffreys, 1939] Harold Jeffreys. *Theory of probability.* Oxford University Press, third edition, 1939.

[Jelinek *et al.*, 1992] Fred Jelinek, John D. Lafferty, and Robert L. Mercer. Basic methods of probabilistic context free grammars. Technical report, Continuous speech recognition group, IBM T.J. Wason Reserch Center, 1992.

[Jobson, 1992] J. D. Jobson. *Applied Multivariate Data Analysis*, volume 2. Springer-Verlag, 1992.

[Johansson and Stenström, 1991] Stig Johansson and Anna-Brita Stenström, editors. *English Computer Corpora: Selected Papers and Research Guide.* Mouton de Gruyter, Berlin, 1991.

[Kamp, 1981] Hans Kamp. A theory of truth and semantic representation. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, editors, *Foraml Methods in the Study of Language*, Mathematical Centre Tracts. Amsterdam, 1981.

[Katz, 1987] Slava Katz. Estimation of probabilities from sparse data for the language model com ponent of a speech recognizer. *IEEE TASSP*, 35(3), 1987.

[Lehnert *et al.*, 1991] Wendy Lehnert, Claire Cardie, D. Fisher, Ellen Riloff, and R. Williams. In *Third Message Understanding Conference (MUC-3)*, pages 223–233, 1991.

[Lehnert *et al.*, 1992] Wendy Lehnert, Claire Cardie, D. Fisher, John McCarthy, Ellen Riloff, and S. Soderland. University of massachusetts: Muc-4 test results and analysis. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 1992.

[Levenshtein,] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and control Theory*, 10(8):707–710. originally published as [Levenshtein, 1965].

[Levenshtein, 1965] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSR*, 163(4):845–848, 1965.

[Lu and Fu, 1977] S. Y. Lu and King Sun Fu. A clustering procedure for syntactic patterns. *IEEE Trans. on Systems, Man, and Cybernetics*, October 1977.

[Meteer *et al.*, 1991] Marie Meteer, Richard Schartz, and Ralph Weischedel. Empirical studies in part of speech labelling. In *Proc. of the 4th DARPA Workshop on Speech and Natural Language*, pages 331–336, 1991.

[muc, 1992] *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, San Mateo, 1992. Morgan Kaufman.

[Novák, 1992] Vilém Novák. *The alternate mathematical model of linguistic semantics and pragmatics*. Number 8 in IFSR international series on systems science and engineering. Plenum Press, New York, 1992.

[Periera *et al.*, 1993a] Fernando Periera, Naftali Tishby, and Lillian Lee. Distribution clustering of english words. In *Proceedings of the 31st Annual Meeting of the ACL*, 1993.

[Periera *et al.*, 1993b] Fernando Periera, Naftali Tishby, and Lillian Lee. Distribution clustering of english words. In *Proceedings of the 31st Annual Meeting of the ACL*, 1993.

[Press *et al.*, 1986] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William Vetterling. *Numerical Recipes*. Cambridge University Press, 1986.

[Pustejovsky and Anick, 1988] James Pustejovsky and Peter Anick. The semantic interpretation of nominals. In Proc. of the 12th International Conference on Computational Linguistics, 1988.

[Pustejovsky and Boguraev, 1993] James Pustejovsky and Branimir Boguraev. Lexical knowledge representation and natural language processing. *Artificial Intelligence*, 1993.

[Pustejovsky, 1991] James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4), 1991.

[Pustejovsky, 1992] James Pustejovsky. The acquisition of lexical semantic knowledge from large corpora. In *Proceedings of the Fifth DARPA Workshop on Speech & Natural Language*, 1992.

[Pustejovsky, 1995] James Pustejovsky. *The Generative Lexicon*. MIT Press, 1995.

[Salton and Schneider, 1983] Gerald Salton and Hans-Jochen Schneider, editors. *Research and Development in Information Retrieval*. Number 146 in Lecture Notes in Computer science. Springer-Verlag, Berlin, 1983.

[Sankoff and Kruskal, 1983] D. Sankoff and J.B. Kruskal, editors. *Time warps, string edits, and macromolecules*. Addison Wesley, Reading, MA, 1983.

[Schütze, 1993] Hinrich Schütze. Part of speech induction from scratch. In *Proceedings of the 31st Annual Meeting of the ACL*, 1993.

[Sellers, 1974] P.H. Sellers. An algorithm for the distance between two finite sequences. *J. Comb. Thy*, A16:253–258, 1974.

[Skakakibara *et al.*, 1993] Yasabumi Skakakibara, Michael Brown, Rebecca C. Underwood, I. Saira Mian, and David Haussler. Stochastic context-free grammars for modeling rna. Technical report, U.C. Santa Cruz, 1993.

[Smadja, 1989] Frank Smadja. Macrocoding the lexicon with co-occurrence knowledge. In *First Int'l Language Acquisition Workshop,* IJCAI 89, 1989.

[Sowa, 1984] John F. Sowa. *Conceptual structures: Information processing in Mind and Machine.* Addison-Wesley, Reading, Mass., 1984.

[Wagner and Fisher, 1974] R. A. Wagner and M. J. Fisher. The string-to-string correction problem. *J. ACM*, 21:168–173, 1974.

[Waterman, 1993] Scott A. Waterman. Structural methods for lexical/semantic patterns. In *Acquisition of Lexical Knowledge from Text, Proceedings of the SIGLEX Workshop*, 1993.

[Waterman, 1995] Scott A. Waterman. Distinguished usage. In Branimir Boguraev and James Pustejovsky, editors, *Corpus Processing for Lexical Acquisition.* MIT Press, 1995.

[Wells, 1957] Rulon S. Wells. Immediate constituents. In Martin Joos, editor, *Readings in Linguistics I.* University of Chicago Press, Chicago, 1957.

[Wilks *et al.*, 1990] Yorick Wilks, Dan Fass, C. M. Gou, J. E. McDonald, T. Plate, and Brian M. Slator. Providing machine tractable dictionary tools. *Machine Translation*, 5, 1990.

[wsj, 1988] The wall street journal, 1988.

[Zernik, 1990] Uri Zernik. Lexical acquisition: Where is the semantics? *Machine Translation*, 5:155–174, 1990.

[Zipf, 1972] George Kingsley Zipf. *Human behavior and the principle of least effort : an introduction to human ecology.* Hafner Pub. Co., 1972. facsimile of 1949 Addison Wesley edn.