THE UNIVERSITY OF WAIKATO

N-gram Models of Agreement in Language

BY

Anthony Clive Smith

A thesis submitted to
The University of Waikato
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in the subject of
Computer Science

Hamilton, New Zealand

June, 2000

Abstract

Conventional n-gram language models are well-established as powerful yet simple mechanisms for characterising language structure when low data complexity is the primary objective. Much of their predictive power can be traced to a relatively small number of common word sequences usually comprised of grammatical terms, and a large number of infrequent word patterns comprised of thematic terms with high mutual information. The drawback for conventional approaches is an exceedingly large number of other n-grams which waste probability mass without making a reciprocal contribution in the formulation of accurate probability estimates.

This thesis describes a simple modification to the n-gram approach which attempts to preserve and enhance the most useful characteristics of conventional models while mitigating their weaknesses by eradicating low utility contexts. If one divides the vocabulary of a language into two broad classes—one comprised solely of contents words (nouns, verbs, adjectives, etc) and the other of grammatical words (determiners, prepositions, modal auxiliaries, etc.)—then language can be viewed as the interlacing of two lexical streams: a content word sequence and a grammatical word sequence. Two words are said to be "super-adjacent" if they are next to each other in one of the two streams.

It is shown that an n-gram model of super-adjacent terms is better able to exploit the high mutual information of close proximity semantic words and the strong syntactic dependencies exhibited in patterns of grammatical words, while many low-utility n-grams that include words from both classes are eliminated. In addition, by reducing regularly inflected words to their base forms and moving inflectional suffixes to the grammatical stream, large numbers of low frequency content bigrams are collapsed into many fewer more general cases, and morphological agreement is made accessible in abstraction. The result is a more compact model that gives better complexity estimates than is possible from the conventional approach.

Acknowledgements

Sincerest thanks and appreciation go to my advisor, Ian Witten, whose support and guidance helped make this thesis (and many other things) possible, and to my co-supervisors, Geoff Holmes and John Cleary, who provided many useful and challenging ideas throughout the period of this research.

I would also like to thank my friends, parents, brother and sisters for their support, and for the tremendous quality of life they create.

Most of all, I would like to thank my wife and children for their immeasurable assistance in the completion of this work; especially for the joy and inspiration they bring to my world daily. This thesis is dedicated to them.

Contents

List of Tables

List of Figures

Chapter 1

Language Modeling

Conventional word-based n-gram models are well-established as powerful and simple mechanisms for characterising language structure when low data complexity is the primary objective. However, they are frequently regarded as linguistically uninteresting in the sense that they fail to capture any sort of grammatical abstraction which might actually form part of a speaker's genuine knowledge of language. Critics object that they play on lexical regularities manifest in the surface form of language without offering any insight into the cognitive phenomena that provide them. There are, to be sure, more complex statistical language models that do attempt to embrace psychologically plausible aspects of grammar, but they are significantly more difficult to formulate and none has yet been proposed that is comprehensive enough to work as a practical account of language in general.

Issues of grammatical perspicuity can be avoided by concentrating solely on the problem of obtaining good complexity measures. N-gram models work very well on these terms, but are nevertheless still subject to some criticism on the grounds that they simply transfer the entropy of language into excessive model size. While there is no doubt that grammatical models can provide a more terse account of language structure, they face a similar counter-objection that language entropy has simply been transferred into the excessive cost of model inference.

This thesis shows how some compromise can be achieved by adapting the basic n-gram approach to incorporate more abstract forms of lexical dependency. The idea is to model the sequential characteristics of grammatical

terms and semantic terms independently. This allows low entropy syntactic patterns to be exploited more effectively while high mutual information in close proximity content words is made more accessible. Furthermore, by reducing inflected words to semantic base forms and treating their inflectional components as free-standing grammatical terms, inflection-agreement dependencies are captured in a more general way. The net result is better entropy estimates from a smaller model without having sacrificed the simplicity, speed and elegance of the basic n-gram method.

The guiding principle of the modeling approach outlined in this thesis is that it is profitable to distinguish between the fundamentally different linguistic functions and relations of open- and closed-class terms. While the model itself is not intended as a theory of learning or grammar per se, many of its procedures aim to exploit aspects of syntax and semantics which emerge through reasoning about the nature of linguistic structure. To that end, it is useful to begin constructing the argument from a brief overview of the philosophy and mathematics of language structure and acquisition.

1.1 Competence-based grammars

Early in the thirteenth century, the Holy Roman Emperor Frederick II of Hohenstaufen undertook a linguistic study wherein newborn infants were given into the sole care of foster-mothers bidden to

> suckle and bathe and wash the children, but in no wise to prattle or speak with them; for he would have learnt whether they would speak the Hebrew language (which had been the first), or Greek, or Latin, or Arabic, or perchance the tongue of their parents of whom they had been born[1].

It would seem that Frederick II had in mind that children came into the world with some sort of innate capacity to speak in a particular language. We may be reasonably sure that, had the experiment been carried through (which, thankfully, it was not), the children would not have spontaneously

[1] from the Chronicle of Salimbene, thirteenth century Italian Franciscan (translated by G. Coulton [34]).

started speaking Hebrew or Arabic; though perhaps they would have instinctively started to use some kind of language if we put any stock in anecdotal claims that twins occasionally invent a language of their own before embracing the language of their parents [11]. For all intents and purposes, however, we regard a specific facility in any one language as something that is acquired—that children must hear a language to learn how to speak it.

Some empirical psychologists of the 1950s adopted a much stronger position by regarding *all* attributes of language development as byproducts of a behavioural stimulus-response system. The developing child hears a word, perceives it to be accompanied with other sensory stimuli and thereafter forms an associative meaning based on that experience. Skinner [96], the champion of radical behaviouralism at the time, explained the acquisition of language as the shaping of a finite single-word grammar through reinforcement by other stimuli. He even went so far as to dispense with the idea that words had any underlying meanings at all, preferring to treat verbal behaviour as the result of "verbal operant conditioning." That is, when a given sequence of words is accompanied by physical reinforcement, as when "give me a drink" results in getting one, or social reinforcement, as when correctly naming an object results in praise and encouragement, then the association is remembered as part of correct grammar. In the absence of such positive reinforcement, as when verbal behaviour is not met with the expected response, or the presence of negative reinforcement, as when an undesirable response accompanies the utterance, then incorrect grammar is negatively reinforced and is forgotten or falls into disuse.

Many linguists at the time had serious misgivings about behavioural models of language acquisition. While the field as a whole was largely preoccupied with developing explicit theories for particular languages, there was a growing awareness that comprehensive grammars and dictionaries were grossly underpowered as a general explanation for the phenomenon of language—particularly when it came to characterising what was then called "the infinite use of finite means." The idea that language was the product of more or less specific cognitive processes had been established in the previous century by the first neurolinguists—Broca, Bouillaud, Wernicke, and so forth—and the highly organised ways in which these mechanisms processed language was

abundantly clear, but exactly how finite mind endowed an infinite capacity for the production and comprehension of language was only just coming to light as the core problem for all of linguistics.

Part of the problem was that, prior to the mid-1950s, linguists lacked sufficient techniques for providing an accurate account of language, but advances in the biological and physical sciences during the first half of the twentieth century provided the necessary formalisms to support revived interest in Frederick II's innateness hypothesis. While other linguists, notably Halliday [52] and Montague [81], proposed generative models to account for the unbounded yet systematic characteristics of language, it is Chomsky's theory of Universal Grammar that has come to form the core assumption underlying nearly all of modern linguistics [23]. The theory states that the basic structures of language are given by innate cognitive mechanisms, and that these are the very mechanisms children use to learn a particular language.

Prior to Chomsky it was widely held that human languages could vary without limit [62], and that any theory of grammar would therefore necessarily be language specific. Chomsky's argument was to the contrary; that the languages of the world have more properties in common than properties which differentiate between them, and that the manners in which they do differ are largely superficial. But providing an explanation for the many similarities observed across the languages of the world was not the key issue in his argument. His ideas were primarily directed against the positive reinforcement view of language learning put forth by proponents of empirical psychology. He argued that learning from "primary linguistic data" (meaning sample utterances) alone could not explain the rapid acquisition of grammatical knowledge and the apparently limitless creativity with which it could be applied in everyday language use. He claimed that these feats could only be achieved if there was some form of generative grammar, with specific characteristics, encased in an inborn biological predisposition. To that end, he began development of a series of grammatical formalisms to characterise universal attributes of language structure [24, 25].

Opponents of universal grammar protested that the innateness hypothesis is fundamentally flawed because it cannot be properly tested, but this objection was convincingly put down with a formal proof by Gold [47] in

1967—and, as Kirsch puts it, "the field of language acquisition has never been quite the same since" [67]. Gold's Theorem proves that it is impossible to learn a correct grammar for an infinite language from positive instances (i.e. sample utterances) alone. The idea underlying his argument is that a learning mechanism can never be certain that the grammar it has inferred will not be disproved by some future example, and that an over-general grammar can never be disproved. Empiricists have been quick to pick up on Gold's observation that the theorem does not apply when the learner has access to negative evidence in the form of an informant (e.g. a parent or teacher) who can tell the learner whether a given input string is well-formed or not, the latter allowing over-general grammars to be eliminated from the hypothesis. Chomsky countered that adult speakers do not in fact routinely provide children with such grammatical judgments, and that children anyway routinely ignore adults who try to correct their speech—claims which are borne out by studies of parent-child dialogue [13]. He further argued that parental speech is so degenerate, deficient and impoverished in the first place that no learner could build an adult language from it, and that if children made assumptions based upon everything their parents said their grammar would be very screwed up indeed [28].

At the heart of the grammar induction problem is the fact that for any one language there exists an infinity of grammars that will provide an account of it. Some neo-empiricists, such as Quine [88], have interpreted this to mean that the problem is one of selection, and have proposed that it can be solved simply by choosing any grammar that is correct with respect to the purpose at hand. Chomsky responds that this misses the point; that what is at issue is not the coverage of the grammar at all but rather its ability to capture genuine elements of linguistic knowledge in a way which rightly says what can and cannot be part of language [27].

As a whole, the school of Chomskyan linguistics might be characterised as one dedicated to the formation of *competence-based* theories of language, typically expressed as a formal system of rules and constraints that explain language structure and provide insight into its manner of acquisition. Opponents argue that grammatical theories of this type are necessarily underdetermined and therefore cannot be said to explain anything [88]. Chomsky

concedes this as true but uninteresting in the sense that all scientific theories are likewise underdetermined—that mathematical formulae which describe the physics of planetary motion or biological theories which describe the evolution of species cannot rightly be thought to explain anything either, but are nonetheless useful tools for reasoning about natural phenomena and predicting future observations, and that this ought to be true for a theory of language. In fact, at times it appears that Chomsky is himself less concerned with complete perspicuity of a grammatical theory than he is with the more general notion that it is at least useful to view aspects of grammar as amenable to some sort of logical formalism [27].

1.2 Performance-based models

In the absence of a sound grammatical theory, it has been necessary for those working on the practical problems of natural language processing to make do with whatever techniques they could devise to realise some progress. Areas such as speech recognition, optical character recognition, part-of-speech tagging, thesaurus building, key-phrase extraction, and so forth have led to the development of a number of statistical techniques able to achieve high levels of success without having to underpin their methods in terms of anything that might specifically be called linguistic knowledge. Such approaches are said to be *performance-based*, where metrics of success are not determined by psychological plausibility of the underlying model, but by its ability to make accurate predictions about the surface properties of language.

By far the most commonly adopted statistical method for achieving good probability estimates of language is the n-gram model [6, 57], a context-based prediction scheme derived from Carnap's proposal for measuring the degree of confirmation for a hypothesis when dealing with the problem of extrapolation from a long sequence of symbols. Carnap states the problem specifically as: given a very long preceding sequence of symbols T constructed from a finite alphabet, what is the probability that it will be followed by the subsequence a? ([21], page 34)

Solomonoff [107] argues that all problems in inductive inference can be expressed in the form of extrapolation of a long sequence of symbols, in that

a sufficiently long T will contain all the information necessary to assign as accurate a probability to a as would be possible from scientific laws. He concedes, however, that in practice it is not possible to evaluate Carnap's probability directly from a mathematical equation, but nevertheless that approximations are possible which are both "qualitatively and quantitatively reasonable." The equation he refers to derives from Bayes' Theorem, and the approximation he has in mind is one which assigns probabilities to finite length T and a. In the case of word-based n-gram models, a is generally restricted to a single vocabulary item, and T is the $n-1$ word history immediately preceding a. The approximation formula can be stated more formally as

$$\Pr[w_k] = \Pr[w_k | w_{k-(n-1)} \ldots w_{k-1}]$$

now commonly referred to as the formula for *sequential maximum likelihood estimation* [19]. Solomonoff makes the very strong claim that a model based upon this form of conditional probability estimation can be made "optimum" in the sense that it would be at least as good as any other conceivable model when it comes to providing an account of a symbolic sequence.

> Other models may devise mechanical explanations of the sequence in terms of the known laws of science, or they may devise empirical mechanisms that optimumly [sic] approximate the behaviour and observations of the man within certain limits. Most of the models that we use to explain the universe around us are based upon laws and informal stochastic relations that are the result of induction using much data that we or others have observed. The induction methods [based on Carnap's probability] are meant to bypass the explicit formulation of scientific laws, and use the data of the past directly to make inductive inferences about specific future events. ([107], page 16)

With respect to natural language, one might interpret Solomonoff's claim to mean that while it is true that language *has* statistically significant characteristics manifest in its surface form, it is by no means the same thing as saying that its underlying mechanisms are themselves probabilistic in nature because any systematic rule-based generative grammar would be expected to

produce highly regular output. But we should nevertheless not allow weak intuitive feelings to dissuade us from having confidence in a Bayesian sequential model that is consistent with the evidence and capable of giving accurate predictions about future observations.

This viewpoint is entirely unsatisfactory from a linguistic point of view, but not just because of its blatant dismissal of the requirement for explicit formulations of psychologically plausible elements. There is a very strong argument against the fundamental modeling power of stochastic techniques which arises in light of *the principle of insufficient reason.* The problem relates to exactly how probabilities are estimated when measuring the degree of confirmation. At any point in time when the model is called upon to assign a probability to a subsequence, a distribution over the possible subsequences must be calculated. In principle, Solomonoff maintains that this distribution should be based upon the number of times each distinct a has been seen to follow T in a string R, where R is sufficiently long that it can be expected to contain a large number of instances of T, and that this distribution becomes increasingly accurate as R approaches infinity. For finite R, it is occasionally necessary to assign probabilities to strings without having seen them, and Solomonoff argues that one should consider all such strings as equiprobable because there is no a priori reason to prefer one over any other. Chomsky argues that this is a significant flaw in all statistical accounts of language in that it inevitably leads to situations where the same degree of likelihood must be assigned to unseen ungrammatical expressions as would be assigned to unseen grammatical ones, and that this is not consistent with the apparent capacity for a mature adult speaker to make accurate grammaticality judgments about novel sentences [23].

Some linguists [85, 49, 74] have argued that this so-called "zero-frequency problem" only undermines the plausibility of statistical language inference with respect to a strictly lexicalist account—meaning that the infinitely plastic combinatorics of words cannot be the actual stuff of grammar formation. But if the generalisation procedures which lead to the construction of a working grammar are directed to appropriate kinds of linguistic abstraction (lexical categories, thematic relations, and so forth), it is at least conceivable that a relatively small characteristic set of expressions would be all that is

necessary for fairly rapid formation of bootstrapping rules. The conjecture still assumes some level of innate linguistic knowledge to get started, but such knowledge need only be a quite modest assumption that semantic and syntactic categories are related in some way. For example, Pinker suggests that a child could infer from context that a word which designates a person, place or thing is a noun, or that a word which designates an action is a verb, or that a word expressing the agent of an action is the subject, and so forth—and that such observations would provide sufficient basis for creating a rudimentary set of simple phrase trees [86]. Once an initial set is learned then more abstract linguistic patterns could be learned through distributional analyses of their appearance in already learned structures.

It seems clear any system that used semantic contexts in such a manner would have to include a rich account of world knowledge, belief systems and the social behaviours required to assign meaning to language. It is equally clear that such accounts are, at least for the time being, completely impossible. If the model cannot detect coincidence of words and meanings—say, for example, by deducing an association between an audible cue and other sensory stimuli—then some other grounds must be given for it to make assumptions about which linguistic phenomena are syntactic and which are semantic. As it happens, there are statistical properties of language which can, to a limited extent, provide such a distinction, and it is these properties that form the basis for the model described in this thesis.

1.3 Towards a unified model

The aim of this thesis is to determine just how a distinction between semantic and syntactic categories might be incorporated into an n-gram based model of language. There are several reasons why this is a worthwhile objective. First, conventional n-gram models have proven to be extraordinarily powerful mechanisms for obtaining low-complexity estimates of language. This is not surprising in light of the fact that typical samples of language do exhibit a great deal of local syntactic regularity in the form of lexical patterns (regardless of whether the underlying generative mechanism corresponds ontologically) and this is exactly the kind of regularity n-gram models excel at

exploiting. Second, n-gram models are extremely simple to build and apply in practical language processing tasks, and their properties are easily analysed. Third, it is difficult to formulate an alternative model that better satisfies the second term of the minimum description length sum: the number of bits required to encode data using the model [89]. This formula is widely used as an immediate measure of the economy of a model, and derives directly from Carnap's probability.

There is one particular aspect of n-gram models that gives them a distinct advantage over other accounts of language (such as phrase structure grammars) and that is the fact that they provide comprehensive coverage—meaning they are not restricted to some subset of sentences with limited syntactic features, but can deal with the infinite variety of real language. Certainly their inability to assign probabilities to unseen contexts is problematic, but other language models suffer from the same problem. The difference is that an n-gram model can recover from an unforseen circumstance simply by adding a new context to its inventory of n-grams, whereas a more abstract structural account may have to be entirely reformulated when confronted with a single novel event.

Despite their strengths, conventional n-gram models do have some serious shortcomings. First, the underlying formalism leads to an exceedingly large model; one which is exponential with respect to the size of the vocabulary and the maximum length context (i.e. the value of n). Because lexical distributions are inherently hyperbolic, so too are the distributions over derivative n-grams, creating a situation where the vast majority of n-grams tend to have very little general utility. For example, Jelinek [57] trained a trigram model on a one and a half million word corpus and then applied it to a second text one fifth as large, only to discover that 25% of the trigrams in the latter had not appeared in the training text. One can deduce from this that, had the roles of the two texts been reversed, 25% of the trigrams in the model would not be able to make a useful contribution to the final complexity estimate for what is presumably a much more representative sample of the language. The implication is that a sizeable portion of the n-gram model inherently corresponds to little more than useless observations.

Second, because n-gram generalisations are restricted to account for ex-

plicit patterns of words, they cannot take direct advantage of higher levels of linguistic abstraction—relationships that reflect more general constraints on how a sequence of words may extend. For example, there is a very real categorial relationship between determiners and nouns to the extent that occurrence of the former signals imminent occurrence of the latter. N-gram models cannot take advantage of this except by gradually collecting a set of exemplars—a set further limited to include only those nounphrases whose length does not exceed n.

Third, there are agreement constraints which give rise to syntactic regularities that are not typically restricted to adjacent words, and thus are not accessible to the n-gram approach. For example, the subject noun and main predicate of a sentence frequently have to agree with respect to grammatical features such as number, person and case. As with categorial dependencies, systematic collation of examples will never allow the model to exploit the underlying linguistic principle, particularly because there is no implicit assumption which could limit the length of context necessary to guarantee inclusion of the two terms involved in the dependency.

There have been attempts to develop language models based on stronger and more general structural formalisms than the n-gram—models capable of assigning probabilities to category-based dependencies [17], recursive structures [22] and feature agreement phenomena [41, 16]. In fact, during the course of research on this thesis several more sophisticated linguistic models were investigated [98–106]. Automatic inference of lexical categories, induction of class-based regular grammars and context-free constituency models, and evaluation of the complexity estimates they provide all realise some degree of success, but are ultimately unsatisfactory because they cannot take advantage of regularities attributable to feature agreement, and they render the problem of model construction at least NP-hard. Automatic acquisition of feature-value grammars and methods by which probabilistic versions could be formulated, trained, and used to assign complexity estimates to language have also been explored, but the construction of such formalisms is at least NP-hard, and anyway the grammars tend to degenerate into lexical mutual information models in a way which negates the very property being generalised—agreement.

Throughout these studies, it has become increasingly apparent that there is another simpler and more fundamental kind of linguistic abstraction available that could be used as the basis for a more robust sequential model of language—a model that unifies lexical, categorial and inflectional regularities. The idea is to distinguish between words whose role is primarily syntactic—the so-called functional categories or "closed-class" words of a language—and those whose role is primarily to provide meaning—the thematic categories or "open-class" words—and model each class separately. The justification for such an approach can be found in a fundamental shift in theoretical ideas about how syntactic structures are formed.

Traditional structuralist accounts of language regard thematic terms as the basic building blocks of syntax. Nouns and verbs (and, in many accounts, adjectives) belonging to the open class are the lexical heads of constituent phrases, and more complex syntactic representations are projections from these terms in accordance with their subcategorisation features [110, 12]. Garrett [45] argued for a complement view of syntax, where the closed-class words—which include determiners, auxiliaries, complementizers, inflectional affixes, and so forth—establish the syntactic framework in accordance with cognitive propositional structures, and thematic categories play a passive role in completing syntactic forms. (Note that prepositions have the unique distinction of being simultaneously closed-class terms and lexical heads in both accounts.) Abney [2] expanded Garrett's ideas into an entire grammatical theory, and since then it has become increasingly common to view the syntax of the world's languages in terms of a characterisation of the inventory and properties of their functional categories [44, 109, 65].

In addition to an increased importance for functional categories in characterisations of syntax, modern theories of language have also shown signs of a general trend away from rigid structuralism towards more relaxed constraint-based accounts. Optimality Theory [4], for example, describes grammar—indeed, even universal grammar—in terms of ranked, violable constraints instead of rules. These constraints regulate structure, making sure everything is present that needs to be in a wellformed utterance, yet allow syntactic exceptions to be handled easily and directly. Similarly, Chomsky's Minimalist Program [30, 70] views grammar as the application of local con-

straints such that derivations are optimised with respect to how well they satisfy constraints of a given item proposed for integration into the structure at each point. The notion that syntax can be described in terms of the optimisation of locally applied constraints suggests increasing compatibility of competence-based and performance-based objectives in modern linguistic theory.

The idea explored in this thesis is whether separate sequential characterisations of functional terms and thematic terms can be parlayed into a smaller, more effective n-gram model than is possible from the conventional n-gram approach. It is important to reiterate that it is not a goal of this thesis to develop a theory of grammar or its acquisition. However, insofar as grammars are thought to be responsible for the regularities in language, a stochastic model that aims to maximise its predictive capacity should benefit from trying to incorporate elements of linguistic theory. This thesis argues that conditional probabilities reflecting lexical, categorial and agreement dependencies can be included within the n-gram paradigm simply by maintaining separate statistics for open-class and closed-class contexts.

1.4 Thesis summary

This thesis describes a super-adjacency model of language: a slightly unconventional word-based n-gram approach that provides better complexity estimates of language from a significantly more compact model than is possible from standard n-gram techniques. The distinguishing feature of a super-adjacency model is that it treats the problem of generalising syntactic phenomena as distinct from the problem of capitalising on semantic relationships.

If one divides the vocabulary of a language into two broad classes—one set comprising contents words (nouns, verbs and adjectives) and the other grammatical words (determiners, prepositions, auxiliary verbs, etc.)—then language can be viewed as the interlacing of two sequences, a content word sequence and a grammatical word sequence. Two words are said to be "super-adjacent" if they are next to each other in one of the two sequences. A super-adjacency model collects separate n-gram statistics for each stream. One set of n-grams capitalises on the syntactic regularities exemplified by patterns

of functional terms while the other set attempts to maximise the mutual information of pairs of content words in close proximity to each other. In addition, inflected content words are lemmatised to base forms, allowing increased cooccurrence of semantically related content words in a considerably smaller model. Inflectional suffixes are moved into the functional stream, allowing slightly deeper contexts to capture agreement relations without entailing a significant penalty in increased model size.

The thesis is organised into seven chapters, each representing a more or less complete treatment of a distinct aspect of n-grams and language. The conclusions drawn from any one chapter form the basis for the argument laid out in the next. This chapter provides a sketch of some of the philosophical and linguistic ideas which establish the motivation for this research. Chapter 2 outlines the formal properties of n-gram models, and provides a detailed analysis of the strengths and weaknesses of conventional word-based n-grams in light of results obtained through experimentation with large samples of English. It shows that while they are good at capturing syntactic structures involving closed-class words, n-grams are unsuitable for utilising lexical adjacencies that involve semantic words.

Chapter 3 describes various lexicalist models that specifically target the high mutual information available in pairs of open-class words regardless of whether they are adjacent or not. It argues that while such lexical attraction models can give good complexity estimates, they are excessive in both model size and processing time, and (like conventional n-grams) the kinds of regularity they uncover are too specific to be useful for general processing tasks.

Chapter 4 details sequential models which focus on exploiting the more general syntactic dependencies that exist between lexical categories. One significant problem for category-based models is their need for accurate part-of-speech tagging prior to training, implying that a linguistic model must already exist before another can be inferred. However, experimentation with the tagging scheme of an unbounded class-based context model shows that conventional part-of-speech labels may be unnecessarily detailed, and that a simple distinction between open- and closed-class words is largely sufficient for preserving many of the categorial relationships that contribute to good

probability estimates.

Chapter 5 introduces the super-adjacency model. Words are assessed as being either open- or closed-class, and each open-class word is replaced by a superclass symbol in the stream of closed-class words. N-gram statistics are garnered from this stream, allowing frequent patterns of grammatical terms, which are of most use in conventional n-gram models, to be preserved, while the plethora of n-grams that involve semantic terms are simplified to more general patterns involving the superclass symbol. A separate n-gram model is constructed for the stream of content words, increasing the incidence of adjacency for pairs of words with high mutual information without a significant loss of category-based prediction offered by the missing grammatical terms.

Chapter 6 argues that most of the utility of mutual information in any strictly word-based account will be watered down in a sea of inflections, where fundamental semantic relationships between base forms of content words are obscured by the diversity of inflectional affixes. A simple technique for detaching inflectional suffixes is described, and experimental results are provided that confirm increased availability of lexical attraction, yet from a significantly smaller model. In addition, by including inflectional suffixes in the set of functional terms, agreement relations can be exploited through the use of deeper closed-class contexts without a significant increase in model size.

Chapter 7 summarises key points from each of the preceding chapters, and consolidates the thesis by highlighting its accomplishments, its significance, and its application to practical language processing tasks. Avenues for future work are also discussed.

Chapter 2

Lexical Markov Models

The fundamental task for a stochastic language model is to predict the next word in a lexical sequence. The major difficulty is that the probability distribution over the set of all possible next words is unknown and must therefore be approximated, typically by garnering statistics from a large training sample. Before this can be done, however, it is incumbent on the designer of the model to decide what information is relevant when estimating a probability.

Lexical Markov models, more commonly referred to as n-grams, assume that the probability of a word is dependent solely on its lexical *history*—the sequence of words immediately preceding the one being predicted. While the assumption is almost certainly wrong, in the sense that many other more abstract forms of linguistic dependency influence language structure, n-grams do surprisingly well at the basic job of predicting the next word.

In this chapter, we provide an overview of word-based n-gram models. We show that, while n-grams are quite suitable for capturing grammatical patterns expressed as sequences of functional terms, they are much less useful for characterising semantic relationships presumed to exist between content words because the incidence of adjacency for specific content word pairs is generally too infrequent. It is argued that semantic relationships are better modeled using techniques that can exploit long distance dependencies, such as the lexical attraction model outlined in Chapter 3 and the super-adjacency model introduced in Chapter 5. In addition, it is claimed that dependencies between function words and content words are not lexical at all, but instead reflect syntactic relationships between their corresponding

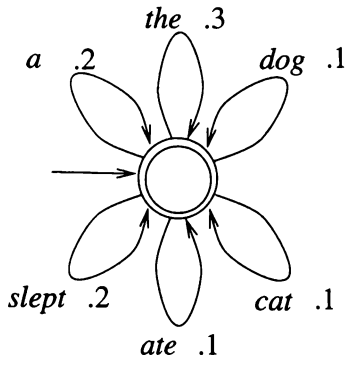


Figure 2.1: A probabilistic single-state automaton for generating words.

grammatical categories—an idea which is explored more fully in Chapter 4.

2.1 Markov models

One can view language as the product of an abstract stochastic process, where words are independent events generated according to some underlying probability distribution. Given such a view, it is possible to model language with a probabilistic finite-state automaton. Consider, as an example, the single-state automaton shown in Figure 2.1 which generates the language $\{a, the, dog, cat, ate, slept\}*$ (i.e. arbitrarily long sequences of any combination of these six words). Given this automaton, we can assign a probability to a lexical sequence simply by calculating the product of the probabilities for each word. For instance, the probability of the sequence "the dog slept" is

$$\begin{array}{lllll} \Pr[\text{“the dog slept”}] = & \Pr[\text{“the”}] & (0.3) & \times \\ & \Pr[\text{“dog”}] & (0.1) & \times \\ & \Pr[\text{“slept”}] & (0.2) & = 0.006 \end{array}$$

In this formulation, the automaton assigns probabilities to all sequences of a given length such that their sum is equal to one. For natural language models it is often desirable to estimate a distribution such that the sum of the probabilities for *all* possible expressions is equal to one—a property that is essential for the model to be appropriately generative. A simple way to do this is to introduce a special *end of sequence* symbol into the vocabulary and

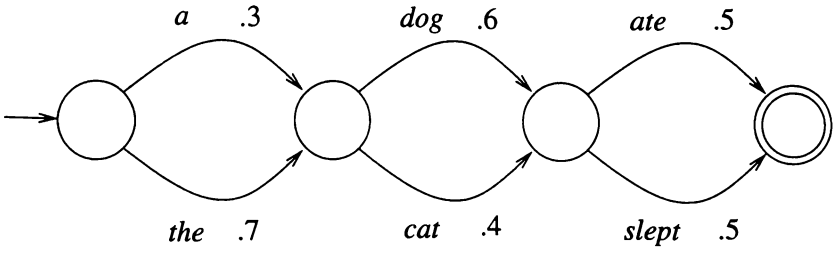


Figure 2.2: A Markov chain for a subset of English sentences.

assign it an appropriate portion of the lexical probability mass. A unique final state is added to the automaton and the transition to it is labeled with the special symbol.

2.1.1 Markov chains

The automaton of Figure 2.1 treats words as independent random events. But words are not generally independent, in the sense that there are presumed to be syntactic and semantic constraints that influence how a linear sequence of words can play out. Thus it is beneficial to introduce structure to the automaton in a way that limits how words can be combined. Consider, for example, the probabilistic finite-state automaton shown in Figure 2.2 which provides an account of just eight English sentences. As before, we can use the automaton to assign probabilities to expressions in the corresponding language by calculating the product of the probabilities for each word, such that the probability of "the dog slept" is now

$$\begin{aligned} \Pr[\text{“the dog slept”}] = \quad & \Pr[\text{“the”}] \quad (0.7) \quad \times \\ & \Pr[\text{“dog”}] \quad (0.6) \quad \times \\ & \Pr[\text{“slept”}] \quad (0.5) \quad = 0.021 \end{aligned}$$

Automata of this type are known as *Markov chains*, named in honour of Alexei Markov who first used them to characterise the statistical properties of Russian texts [77].

Probabilities for words in a Markov chain are not entirely independent, but are conditional on the current state. Exactly what information is encoded in the state is given by the automaton's structure and is characterised in terms of the path (or paths) through the Markov chain leading to the state.

For example, the probability for "dog" in the automaton of Figure 2.2 is dependent (at least) on the immediately preceding word being either "a" or "the". In this respect, state information effectively divides a vocabulary into equivalence classes, where each class is defined as a set of words sharing a common lexical history.

2.1.2 Markov approximations

It is not feasible to construct Markov chains with states that correspond to every discrete lexical history. In fact, it is not desirable because almost every sentence in even a very large language sample is unique, suggesting that corresponding equivalence classes are unlikely ever to be called upon to predict a future event.

More generally, lexical dependencies tend not to extend over great distances. It is, for example, unlikely that the first word of this thesis exerts much influence on the first word of this sentence. Syntactic constraints tend to be sentence bound, and are often restricted just to the current constituent phrase—as when a determiner predicts the noun within the same nounphrase, but says nothing about nouns in any other phrase. And, while semantic constraints may carry on over a long sequence of language dealing with a specific topic, recent semantic terms alone are likely to be sufficient for maintaining the relevant dependency. Therefore, we might tacitly assume that any distant dependency is sufficiently weak that ignoring it will not affect our ability to predict the next word. A reasonable model of language is thus one that treats words as conditionally independent of past linguistic events *given* the local context.

It is practical to make a *Markov assumption* that only a finite amount of the immediate lexical history is needed to make reasonably accurate predictions about what word will occur next. Specifically, a k-th order *Markov approximation* assumes the probability of the next word is dependent solely on the previous k words. More formally,

$$\Pr[W_i = w_i] = \Pr[W_i = w_i | W_{i-k} = w_{i-k}, W_{i-k+1} = w_{i-k+1}, \ldots, W_{i-1} = w_{i-1}]$$

where $\Pr[W_i = w_i]$ is the probability of the word w_i in the equivalence class W_i [76].

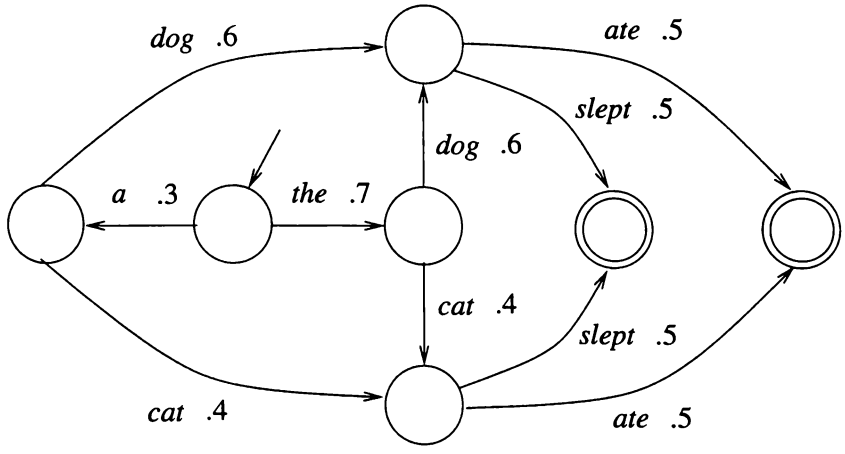


Figure 2.3: An order-1 context model for a subset of English sentences.

2.1.3 N-gram models

It is relatively straightforward to design probabilistic finite-state automata based on Markov approximations. The model in Figure 2.3, for example, corresponds to a first-order Markov approximation for the same language covered by the automaton in Figure 2.2. But, whereas the previous model must assign the same probability to "dog" regardless of whether "a" or "the" came immediately before it, the new model encodes the previous word exactly and can therefore assign different conditional probabilities to "dog" for its two possible single-word contexts. (Note that the model in Figure 2.1 can be said to model a *zero*-order Markov approximation because no preceding word influences the probability assigned to the next.)

A model in which every state defines an equivalence class for all words whose histories are exactly the same $n-1$ words is called an *n-gram model*—also referred to in statistical literature as an "(n-1)-th order Markov model" or an "order-(n-1) context model". Such models are highly effective at exploiting regular local syntactic constructs in language, and can provide reasonably accurate lexical predictions from even quite limited prior context. For example, the word "the" occurs in the Brown Corpus with probability 0.062 under a zero-order Markov approximation, but with probability 0.27 given a first-order Markov approximation when the word "of" occurs immediately prior to it, and with probability 0.58 following the words "one of"

from a second-order Markov approximation.

2.1.4 Data sparseness

There is, in principle, no limit to the amount of context that can be used when predicting the next word, and in general one expects longer contexts to provide more accurate estimates of the lexical distribution. But the cost for greater context is an increase in model size that is exponential with the order of the model, such that a comprehensive k-th order Markov model for a language with vocabulary size V has V^k equivalence classes. Given a typical English vocabulary with, say, twenty thousand words, a third-order Markov model would have to maintain statistics for almost 10^{13} separate n-grams.

In practice, Markov model states are constructed only for contexts actually observed in a training sample of the language. Even quite large samples tend not to contain all possible k word patterns and, as a consequence, models are usually quite a bit smaller than their potential worst case because unseen contexts can be treated uniformly as one equivalence class (i.e. an unused state). As the order of the model increases, the proportion of possible n-grams that are actually observed diminishes quickly, to the point where the absolute number of n-grams in the model eventually starts to decrease. Consider the extreme case where an k-th order Markov model for a language sample of length $k + 1$ has just one n-gram.

Recall, however, that the whole point of using a Markov approximation is to avoid having too many equivalence classes that never predict a future event. Patterns of more than four or five words are quite rare even in very large language samples, meaning only a few of the possible n-grams in an order-five Markov model would ever be observed. More importantly, because lexical distributions are inherently hyperbolic, so are corresponding n-gram distributions, thus higher-order Markov approximations tend to produce very few equivalence classes that are useful for predicting words.

Accurate predictions only become possible when a model has seen a sufficiently large language sample to garner reliable statistics. As the order of the model increases linearly, the amount of training data required for it to support useful and accurate predictions increases exponentially. To avoid this problem of data sparseness, most practical models are limited to first-order

or second-order Markov approximations—commonly called bigrams and trigrams respectively [57].

Despite using very little lexical history, low-order context models are still able to deliver exceptionally good probability estimates in comparison to just about any other stochastic language model. And precisely how and where their predictive gains are made is of specific interest for this thesis.

2.1.5 Typical language

Before exploring the performance characteristics of various conventional n-gram models, it is expedient to describe the language sample used for all experiments in this thesis.

The *Brown Corpus* was compiled in the late 1960s in an attempt to create a single, large, representative sample of American English for language research [68]. We chose to use this sample for studying the behaviour of various statistical language models, not so much because it is "typical"—a claim we do not wish to defend at all—but because it is probably the most widely used corpus available, and its characteristics are well known. These features allow for the results and analyses outlined in this thesis to be challenged or confirmed more easily than if we had used corpora that are less readily available.

Because this research focuses specifically on stochastic characterisations of dependencies between words, however, it was deemed useful to make a few small changes to the Brown Corpus to remove irrelevant format details that might obscure fundamental lexical properties. Prior to training, the corpus was downcased (i.e. upper case letters changed to lower case), and all sentence-internal punctuation was removed (i.e. commas, colons, quotation marks, etc.) except apostrophes. End of sentence punctuation marks (i.e. periods, question marks and exclamation points) were retained as sentence boundary markers, and as such are incorporated into the relevant statistics as individual vocabulary items. The result of preprocessing is a corpus of 51,279 sentences comprised of 1,065,795 tokens with a vocabulary size of 44,519 different words.

2.1.6 Information and uncertainty

Lexical probabilities are typically quite small and, given that a stochastic sequence model calculates the probability of a sentence as the product of the probabilities of its words, the resulting value is often so small that it is cumbersome to express as a real number, and comparison of the probabilities of two sentences can be difficult to appreciate. It has become common practice to express expectations about a random linguistic event in terms of the number of bits required to specify the outcome given a particular model [89].

The idea originates from communications theory, where the information content of a message is measured directly from its probability given the expectations of the recipient [95]. When the recipient knows with perfect certainty what a message will be, then the information content of the message is effectively zero and it need not be sent. If, however, the expectations of the recipient lead to some uncertainty about specific aspects of the message, some information must be included to help disambiguate what is intended by the sender from all possible meanings the recipient might reasonably assume.

In this respect the quantity of meaning in a message is in a very real sense exactly the same as the amount of disambiguation information it requires. One can view the situation as a kind of *twenty questions* game, where the uncertain recipient seeks clarification by asking "do you mean this?" and the sender responds with "yes" or "no". Relatively transparent meanings generally require fewer questions than more obscure interpretations, and communications theory uses this idea to assign a precise value to the amount of ambiguity (i.e. meaning) in a message. More plainly, the number of questions that must be asked in order to assign the correct meaning to a message is a direct measure of its uncertainty.

How many questions must be asked, however, depends a lot on the expectations of the recipient. Some questions are better than others and will lead the recipient to the correct interpretation more quickly than less prudent ones. But good questions will only be asked if the recipient makes valid assumptions about what information is most relevant for rapid disambiguation. Shannon [94] shows that the number of questions needed is directly related to the probability assigned to the message according to the expectations of

the recipient, such that it is equal to the negative base-2 logarithm of that probability. Formally, given a message with probability p, its uncertainty (i.e. information content) is equal to $-\log_2 p$.

It is perhaps easier to understand the relationship between probability and uncertainty with the standard example of a coin toss. Given a fair coin, the probability that the outcome of a single toss is "heads" is equal to one-half. The uncertainty of the outcome is thus equal to $-\log_2 1/2 = 1$, and this corresponds to the number of questions one would have to ask to determine whether the concealed outcome of a toss was actually heads.

2.1.7 Complexity and entropy

Complexity-based induction theory has a more general way of talking about uncertainty [108]. Given a theory T that explains some examples E, the minimum number of bits required to encode E using T is the complexity of E with respect to T [33]. Complexity of examples given a theory is exactly the same thing as uncertainty of a message given expectations, but the former is perhaps a more natural way of looking at the performance of a stochastic language model. A good theory will encode examples more efficiently than a bad theory; thus complexity is a useful way of comparing the relative soundness of two competing theories. Similarly, a good language model will assign probabilities to words more accurately than a poor model, thus we can compare the performance of two models by calculating the complexity of the same language sample with respect to each model.

Another important metric for evaluating a model's characteristic behaviour is the *average* uncertainty it entails about the outcome of a random linguistic event—a measure more commonly referred to as the *entropy* of a random variable. Shannon [94] proposed that the entropy of an event with probability p can be calculated as $-p\log_2 p$, and this formula is now often taken to be the accepted definition of entropy.

Entropy is an important analytical measure of how good a model is at predicting future events. Consider the coin toss example again. This time, instead of asking whether the outcome of the toss is heads, we simply assume that it is heads and rely on the person who tossed the coin to correct us when we are wrong. On average, we expect to be corrected only half the time, thus

the amount of disambiguation information we are likely to receive over the long haul is just half of what we need when we do not make asumptions about the outcome.

The terms entropy, complexity, uncertainty and information content are used throughout this thesis, and most often they are meant to be references to more or less the same thing: the negative log likelihood of a random event. If the reader prefers a less mathematical connotation, we recommend they keep just one general principle in mind: high probabilities, low complexity and low entropy estimates are all desirable when seeking a good model of language.

2.2 Unigram models

A unigram model is a zero-order Markov model that predicts the next word in isolation based on the global distribution for the entire vocabulary as observed in a finite sample of language. That is, the probability of a word is proportionate to its frequency with respect to the total number of words observed. Despite their simplicity, unigrams allow a number of useful observations.

Figure 2.4 plots the unigram complexity estimates for a random sentence taken from the Brown Corpus. Insofar as the sentence is 'typical', the graph implies that most of the complexity of the expression originates from the highly semantic terms (like nouns and adjectives), while terms whose role is predominantly grammatical (like determiners and prepositions) are substantially easier to predict. A straightforward explanation for this is simply that most grammatical terms occur much more frequently than most semantic terms; a property that unigram models specifically aim to capitalise upon. But exactly why grammatical terms occur so much more frequently is better understood through an account of their role and behaviour in terms of linguistic and communications theories.

2.2.1 Complexity of semantic categories

Entropy estimates given by unigram models conform with basic intuitions about the distribution of meaning within the words of English expressions—

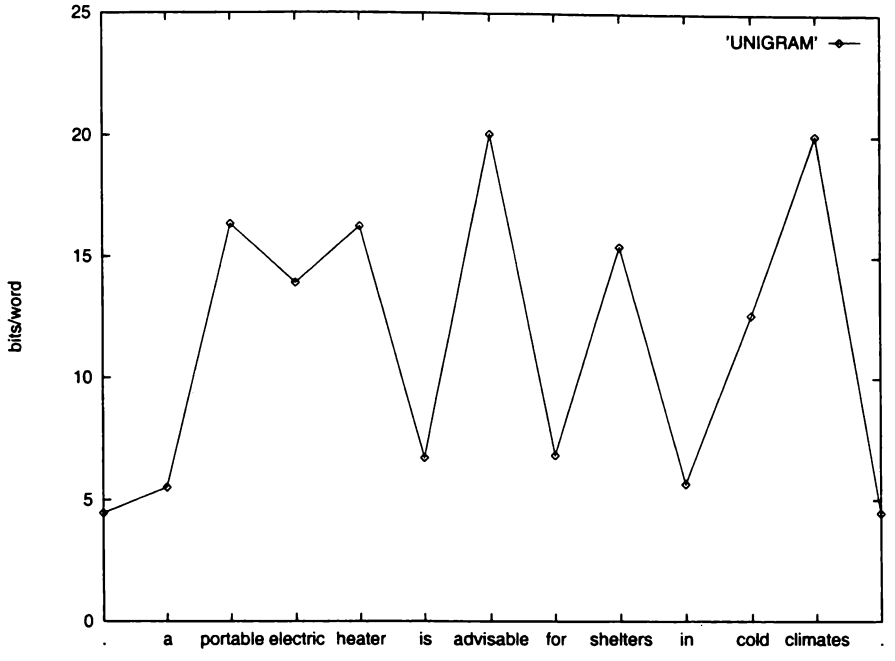


Figure 2.4: Unigram complexity estimates for a typical Brown Corpus sentence.

at least with respect to a strictly lexical account. That is, given that the bulk of meaning for an expression is carried by its semantic terms, we expect higher complexity estimates to be associated with them than we do for grammatical terms. Indeed, Table 2.1 lists fifty of the 17,879 *unique instance* words in the Brown Corpus, and the fact that these are all semantic terms with a probability less than one in a million is consistent with the expectation that meaning in language entails high uncertainty for the words.

Certainly grammatical terms carry some meaning, but such meaning is rather more abstract than it is for semantic terms. For example, in the expression "the lion killed the tiger", virtually all of the meaning is embodied in the nouns and verb: "lion", "killed" and "tiger". But, by themselves, these semantic terms fail to make more subtle aspects of the intention clear. Consider, for comparison, the expression "the lion was killed by the tiger". It has the same semantic terms in the same order, but the intended thematic roles are reversed such that the subject of the sentence, previously serving as the agent of the action, is now the patient. This reversal of thematic roles

acclaims	acoustics	operationally	turnaround	wetlands
accommodated	acquiesced	opportunism	twirled	wiggle
accompanist	acrobats	preceeded	typographic	womanhood
accompanists	hydraulics	presupposition	unacknowledged	worksheet
accountable	icicle	robustness	unthinking	wretchedness
accountants	lunchroom	striptease	uprooted	wronged
acculturated	milestones	subsistent	vocalization	yearn
ackerly	mountaineering	summarization	voiceless	yellowish
acorns	moustache	sundials	warmongering	zealot
acoustic	mustering	transferral	weeded	zombie

Table 2.1: Fifty single instance words from the Brown Corpus.

is signaled by additional syntactic cues given by the grammatical words. As rich in meaning as nouns and verbs are, they cannot (at least in English) embody subtly important aspects of nuance.

2.2.2 Complexity of grammatical categories

The relatively specific semantics of nouns, verbs and adjectives are intimately related to topic, world knowledge, environment, and other aspects of discourse, thus any particular content word is generally infrequent in large representative samples of language. Conversely, grammatical terms provide additional linguistic information needed to clarify the intended meaning of the content words. Determiners, for example, supply quantification and specification of referent for nouns; auxiliaries add mood and tense to verbs; and prepositions carry time and space juxtapositions (among other things) for subjects and objects. Such subtle characteristics of meaning are among the general requirements of all speech acts, and thus the words that provide them are necessarily quite frequent in just about any language sample.

Table 2.2 lists the fifty most frequent terms in the Brown Corpus, along with their counts, and all are grammatical words (end-of-sentence markers excepted) from the *closed class*—a set so-called because its membership is not open to the introduction of new terms. Closed-class words are also called "function words" because their function in language is to signal certain forms of syntactic structure (such as relative clauses, verbal complements and questions) and to introduce subtle semantic notions (such as anaphora, possessive

69935	the	9542	he	5240	I	3618	they	2670	their
48427	.	9481	for	5143	this	3560	which	2643	we
36389	of	8757	it	5131	had	3410	one	2619	him
28854	and	7285	with	4612	not	3285	you	2472	been
26138	to	7248	as	4387	are	3284	were	2438	has
23421	a	6996	his	4380	but	3050	all	2330	when
21337	in	6755	on	4367	from	3037	her	2250	who
10588	that	6375	be	4207	or	2859	she	2243	will
10096	is	5378	at	3938	have	2724	there	2231	?
9815	was	5321	by	3738	an	2720	would	2216	no

Table 2.2: Fifty most frequent words from the Brown Corpus.

and future tense). There are approximately 500 function words in English [20], and they have a number of significant linguistic properties—the most salient of which (when language is viewed as a statistical process) is that most function words are extremely frequent. In fact, if Table 2.2 was extended to include the next fifty most frequent words from the Brown Corpus, only four ("said", "man", "time" and "years") would not be function words. This statistical attribute proves particularly useful when it is necessary to determine which words are likely from the closed class and which are not—a determination required by several models outlined later in this thesis.

In the absence of grammatical terms, English expressions have a kind of default assignment for thematic roles. Given an expression like "lions kill tigers" it is clear that the subject of the sentence is also the agent of the main verb because that is the convention of the language. Thus it is that word order does make some contribution to clarifying subtle semantic differences. In addition, the mere presence of grammatical terms does not necessarily elicit a distinction in semantic nuance because they must be properly juxtaposed with the semantic terms they are meant to complement. A determiner precedes the noun it qualifies, a modal auxiliary precedes the verb whose mood is to be altered, and a preposition precedes the adjunct nounphrase to which it applies. In addition, only adjectives (generally speaking) may stand between a determiner and its referent noun, only certain adverbs may stand between a modal and its associated verb, and nothing may stand between a preposition and its associated nounphrase. So it is said that English is *right-*

ward selecting in that functional categories exhibit selectional properties that project rightward over the immediately following complement structure [83].

Determiners appear exclusively in nounphrases, and whenever a determiner appears it marks the onset of a nounphrase [1]—meaning that the occurrence of a determiner makes the imminent appearance of a noun a virtual certainty. This suggests a genuine linguistic relationship between determiners and nouns—and similar relationships can be supposed between modals and verbs, and other lexical pairs. Such causative associations imply the availability of lexical presentiments which can be exploited for predictive purposes. More plainly, where a unigram model assigns a probability to, say, a particular noun based on that noun's frequency with respect to the total number of words observed in a language sample, a higher-order model which has access to the prior context of a determiner can improve its estimate by distorting the distribution for possible next words in favour of nouns.

2.3 Bigrams and higher-order models

A bigram model is a first-order Markov model that establishes a distribution over the set of possible next words based on the context of the previous word. The probability of a word is therefore proportionate to the frequency with which it has been seen to occur immediately following the previous word, with respect to the total number of times the previous word has been seen.

Figure 2.5 graphs a comparison between the unigram (top line) and bigram (bottom line) complexity estimates for the Brown Corpus sample sentence given in the previous section. As expected, the graph shows a general and substantial improvement in the predictive power of a bigram lexical model over that of the unigram model, and the gap between the two lines refects the savings in the complexity estimates. Just how the overall savings are distributed over the sentence offers some clues about which aspects of language are being exploited by the bigram technique.

2.3.1 Spurious semantic savings

Table 2.3 itemises the complexity estimates pictured in Figure 2.5, listing bigrams in descending order according to the bit savings each offers over a

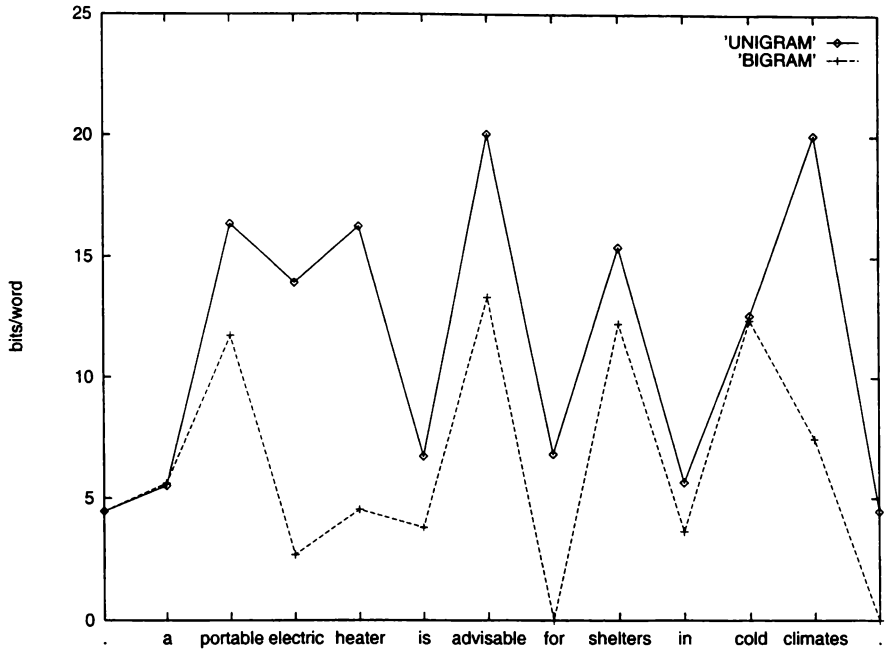


Figure 2.5: Unigram and bigram complexity estimates for a sample sentence.

unigram model when it comes to predicting the next word. For example, the second row of the table indicates that the word "heater" occurs 14 times in isolation and 3 times immediately following "electric". The complexity of "heater" in a unigram model of the Brown Corpus is $-\log_2 14/1,065,795 \approx 16.22$, but is only $-\log_2 3/70 \approx 4.54$ following "electric" in a bigram model, giving a net savings of about $16.22 - 4.54 = 11.68$ bits. Note that the last row of the table shows the probability of "a" following a fullstop is actually a bit lower than its independent probability, thus its complexity estimate is slightly higher in the bigram model.

The table shows that the most significant savings occur when one semantic term is able to predict another. Statistically speaking, this result is actually due to the relative infrequency of the semantic word providing the context. If a particular symbol does not occur very often then the set of possible next symbols must be quite small. However, if this was all there was to it then one might not expect to see quite such substantial savings. For example, 49 different words occur immediately following "electric" in the Brown Corpus. If the distribution over these 49 words were uniform then the

w_i	freq. w_i	w_{i-1}	freq. $w_{i-1}w_i$	$-\log_2$ $\Pr[w_i]$	$-\log_2$ $\Pr[w_i\|w_{i-1}]$	bit savings
climates	1	cold	1	20.02	7.44	12.58
heater	14	electric	3	16.22	4.54	11.68
electric	70	portable	2	13.89	2.70	11.19
for	9481	advisable	1	6.81	0	6.81
advisable	1	is	1	20.02	13.30	6.72
portable	13	a	7	16.32	11.71	4.61
.	48428	climates	1	4.46	0	4.46
shelters	25	for	2	15.38	12.21	3.17
is	10096	heater	1	6.72	3.81	2.91
in	21337	shelters	2	5.64	3.64	2.00
cold	174	in	4	12.58	12.38	0.20
a	23421	.	999	5.51	5.60	-0.09

Table 2.3: Log likelihood gains for bigram model over unigram model.

contextual frequency of each word in a bigram model would be $70/49 \approx 1.4$, and the negative log likelihood of each word would be $-\log 1.4/70 \approx 5.6$. In actual fact, just 14 of the words occurring after "electric" account for half of the observations, and "heater" is the third most frequent with a bigram complexity of only 4.5.

The word "heater" occurs three times after "electric" in the Brown Corpus. Given that the unigram probability for "heater" in the Brown Corpus is just $14/1,065,795$, the probability that any word of such frequency would be chosen at least three times in 70 independent selections is exceedingly small. More to the point, given that $\Pr[\text{"heater"}] \ll \Pr[\text{"heater"}|\text{"electric"}]$, we can be virtually certain that the independence assumption is false, and that the likelihood of "heater" is positively influenced by the prior occurrence of "electric".

Table 2.4 lists thirty of the best predicting bigrams from the Brown Corpus, and it is consistent with the proposal that bigrams comprised of two semantic terms buy the greatest savings in complexity estimates. In fact, all of the bigrams listed reduce the negative log likelihood of the second word by over twenty bits. But the table is misleading in that all of these bigrams occur just once in the corpus. In fact, more than two thirds of the vocabulary of bigrams in the corpus is comprised of single-instance word pairs, thus the

loosest possible	ambitiously coveting	halfhearted acclamation
lustful stares	unfunnily sarcastic	transatlantic jetliners
viva voce	uneducated newlywed	thrombosed hemorrhoids
jolting tackles	hypodermic needle	breezy clotheslines
burglar alarms	transistor oscillator	horselike balkiness
acrid stench	pestilent seducer	scurrilous underhandedness
highschool dramatics	budding womanhood	forefingers darting
gnarled talons	stoutly replying	petulant admonition
plantain lilies	lunchroom suppers	raindrops pattered
prayerful forepaws	yellowed prayerbooks	scathingly condemnatory

Table 2.4: Thirty of the best predicting bigrams from the Brown Corpus.

contribution each makes (in isolation) to net savings for the entire corpus is virtually nothing.

This distortion of bigram utility arises in part because we are using *post hoc* probabilities derived from sparse data. The tremendous savings obtained by predicting "possible" based on the previous word "loosest", for example, is achieved because the independent probability of "possible" in the training sample is just $373/1,065,795 \approx 0.0003$ but is $1/1 = 1$ using a first-order Markov approximation, given this particular context. While it is true that the word "possible" always follows the word "loosest" in the Brown Corpus, it is not true for English in general. Had the training set been bigger it may have been a little more representative of English, and complexity estimates from some of these rare bigrams might better reflect their general utility for predicting future events. We discuss the problem of using posterior probabilities for model evaluation in the next section, but we can still get a better idea of the true utility of an n-gram from such raw figures by considering its average expected savings—its entropy.

2.3.2 Bigram utility

To gain a better understanding about those properties of bigram models which are more generally useful for language modeling, the expected savings attributable to a given bigram must be moderated in terms of its likelihood. A bigram that offers tremendous savings but hardly ever occurs is generally

savings	bigram	savings	bigram
0.019454	of the	0.0125541	in the
0.0060538	on the	0.00602654	it is
0.00585149	to be	0.00513457	it was
0.00450018	had been	0.00405244	united states
0.00395765	have been	0.00389343	he was
0.00382503	he had	0.00369977	has been
0.00364831	at the	0.00352614	to the
0.00321004	from the	0.00319525	will be
0.00313095	would be	0.00287923	for the
0.00287317	did not	0.00279006	can be
0.00275826	more than	0.00274467	new york
0.00260659	may be	0.00260316	by the
0.0025805	as a	0.00255915	with the
0.00254367	there was	0.00239885	there is
0.00235523	the same	0.00227233	with a

Table 2.5: The thirty most useful bigrams in the Brown Corpus.

less useful than a very frequent bigram that provides relatively small instantaneous savings. A better estimate of the general utility of a bigram can be obtained using the general entropy formula $-p \times \log p$, where the average savings given by a word pair $w_{i-1}w_i$ is calculated as the difference between the complexity of w_i in a unigram model and its complexity in a bigram model *times* the probability of the bigram, or

$$\Pr[w_{i-1}w_i] \times -(\log_2 \Pr[w_i] - \log_2 \Pr[w_i|w_{i-1}]).$$

Table 2.5 lists the thirty most useful bigrams from the Brown Corpus, along with their expected entropy savings. In stark contrast to the preliminary findings presented in Table 2.4, where pairs of semantic terms offered the most savings, Table 2.5 indicates that the best savings are actually associated with pairs of grammatical terms. This is not universally true, however, as "united states" and "new york" are clearly semantic bigrams. It may be tempting then simply to attribute the low entropy estimates to the relative high frequency of the bigrams, but this does not hold in all cases. For example, the bigram "with the" appears over 1500 times in the Brown Corpus, while "new york" occurs fewer than 300 times, yet the latter gives better entropy savings. Moreover, there are over a hundred bigrams that occur more

frequently than "new york" which do not offer more accurate predictions.

One explanation for the results may be that there are two kinds of lexical relationship providing useful bigram complexity estimates. The first is a purely syntactic regularity wrought from basic grammatical requirements. Bigrams such as "in the", "on the", "at the", "from the", "by the", "with the", "with a", "of the", "for the" and "to the" typically introduce adjunct phrases (though the statistics will include instances where prepositions are being used as verb particles), which are common syntactic constructs in any sample of English. Further, "had been", "have been", "has been", "will be", "would be", "can be", "may be", "to be" and "did not" are auxiliary and modal verb combinations required for perfect tenses and the tensing of defective verbs. Both of these are also highly utilised grammatical forms. Thus it appears that the bigram model actually captures regularities relating to bona fide grammatical requirements, rather than any sort of intrinsic relationship stemming from pure lexical semantics.

The second kind of lexical relationship captured in low entropy bigrams is one that relates to complex noun structures. Almost all of the most useful bigrams in the Brown Corpus not comprised of two grammatical terms are pairs of semantic terms tightly coupled due to a convention in English nomenclature. For example, following on down a more comprehensive list of low-complexity bigrams, after "new york" one will find "rhode island", "peace corps", "per cent", "los angeles", "high school", "white house", "fiscal year", "nineteenth century", "world war", "united nations", "general motors", "president kennedy", and so on. To say that the low entropy of these word pairs is due to their strong semantic relationships is somewhat of an overstatement, as they are perhaps more rightly viewed as single terms that are epiphenomenally bigrams.

2.3.3 Characterising bigram utility

Close examination of entropy savings available from bigram models indicates that the vast majority of bigrams offer little better general predictive accuracy than what is available from a unigram model. Figure 2.6 plots the expected bit savings of the 1000 most useful bigrams found in the Brown Corpus. The evidence indicates that, beyond the first few hundred bigrams,

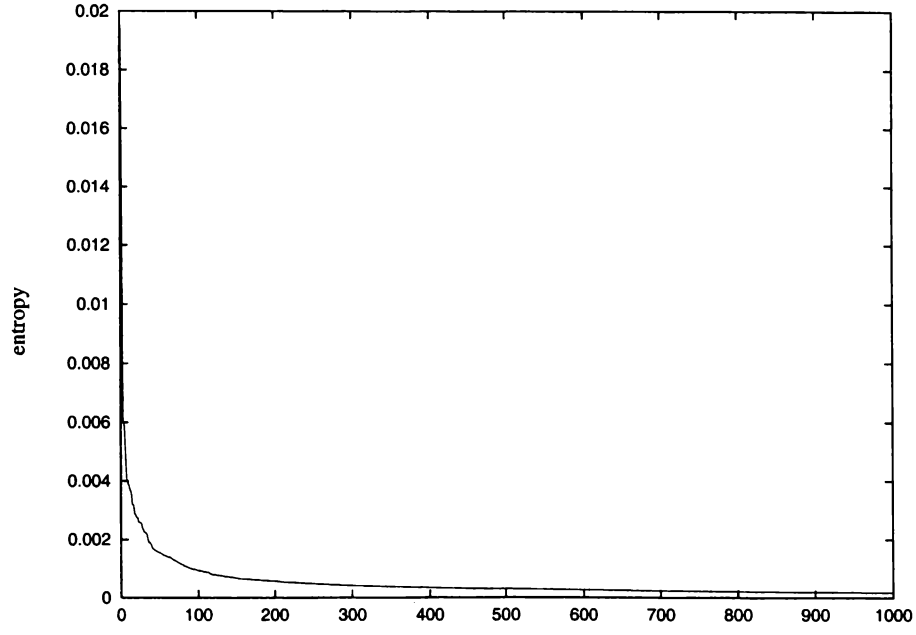


Figure 2.6: Expected savings for the 1000 most useful Brown Corpus bigrams.

contributions made by individual bigrams to reducing the overall complexity estimate for the corpus are largely insignificant. In fact, if the x-axis was extended to incorporate the remaining 430,705 bigrams in the corpus, the savings offered by the best bigrams would become obfuscated and the overall graph would appear to be a flat line at zero.

Despite this ostensively poor result, and despite the significant entropy savings made possible by the top few hundred bigrams, most of the overall savings offered by bigram models can still be traced to this extraordinarily large number of rare word pairs. For example, single-instance bigrams account for 84.5% of the total bit-savings for the Brown Corpus, bigrams occurring once or twice provide 92.5% of the savings, and those observed five times or less account for almost 97.5% of the total savings. But these low-utility bigrams are by no means uniform. When one examines the instantaneous savings of infrequent bigrams, noticeable trends emerge which highlight a significant weakness of the modeling paradigm. While some Brown Corpus bigrams can save as many as twenty bits over a unigram model when encoding a word, there are very many others that save nothing at all or,

the administrator	the execution	the placement
the apprentice	the explanations	the policemen
the bankers	the expressions	the protestants
the bees	the federation	the refusal
the blast	the fox	the rolls
the bottles	the memories	the selections
the businessmen	the misery	the stature
the celebration	the onion	the texture
the creature	the optimism	the vapor
the divisions	the picnic	the writings

Table 2.6: Thirty bigrams that save just one bit over unigrams.

worse still, produce even less economical encodings for words than the unigram model—a situation which arises when the frequency with which a given word is observed following a particular context is less than its frequency with respect to the entire corpus.

Bigrams that give large instantaneous savings are almost always pairs of content terms—an observation one could reasonably attribute to a semantic relationship shared between them. Bigrams that give very low or negative savings, however, are characteristically ones that pair one content word with one grammatical word. Why these pairings lead to such poor joint probabilities is explained by the nature of the relationship between adjacent content and grammatical terms—a relationship that is far more syntactic than it is semantic.

When a determiner is observed, the onset of a nounphrase is signaled and the probability of a noun appearing soon should increase. But the context afforded by the determiner does not provide much selectional power beyond this. Given that nouns typically constitute nearly half of all vocabulary items, knowing that a noun is imminent can only save about one bit when it comes to encoding any particular noun. Table 2.6, for example, lists thirty bigrams where the context provided by "the" saves just one bit over the unigram complexity estimate for the following noun. Certainly the determiner may include additional feature constraints, as when indefinite articles select singular form countable nouns, but such constraints can only halve or quarter the set of nouns and thus will only save another bit or two over a unigram

code. The bulk of the encoding cost for a noun is directly attributable to its rarity (i.e. its information content) and any savings that arise from the context of a preceding determiner are negligible given how many different nouns occur following determiners.

Similarly, once a noun has appeared there is considerable justification for decreasing the probability of another noun occurring immediately following, thus about one bit may be saved. When the next word is another content term (usually a verb) it is highly likely to exhibit a strong semantic association with the preceding noun and considerable savings result. That is, the semantics of a noun often create strong selectional constraints over the next semantic term, and this translates into large savings when encoding that term. If, however, the next word is a grammatical term then the semantic relationship is often too weak to distort the distribution very much in favour of any particular function word. Given that function words constitute about 50% of the tokens in a typical language sample, knowing that a function word is imminent does not produce much more than about one bit of savings over a unigram based encoding.

It appears then that the majority of entropy gains for bigram models result from their ability to capture grammatical structures embodied in a small number of patterns comprised exclusively of functional terms, along with the combined effect of large instantaneous savings from copious infrequent content word pairs, but that vast numbers of other bigrams are all but useless for reducing estimates of language complexity. This tendency towards underutilisation of specific lexical patterns implies that bigrams are not an effective mechanism for capturing language structure in general—that much of the model's time and space is needlessly wasted keeping track of an overwhelmingly large number of uninteresting observations of negligible value.

2.3.4 Up from bigrams

Increasing the order of an n-gram model increases the amount of context available for predicting the next symbol and probability estimates generally improve accordingly. As noted earlier, however, the cost for such improvement is very many more parameters in every distribution. Leaving aside the problem of model size for the moment, it is useful to analyse some of the

brightly colored lithographs	intellectual sterility spruced
discovered ancient yoga	jet fighters strafing
entitles conscientious objectors	jubilantly reunited bunkmates
face shouted senselessly	looked grotesquely unshaven
fourteen glamorous schoolgirls	loose indian insurrections
glorious silver punchbowl	lower motor neuron
graduate students abstractors	mouth grinning trustfully
great intellectual coherence	naked hair queued
hate minute polemics	numerous times rebuked
helping attract fairgoers	occasionally introduced smel
hoarse old mastiff	overdeveloped lower jawbone
hundred hidden malevolencies	payment vouchers certifying
illegal gambling dens	political restraint subdues
independent states balkanizing	remained largely unexamined
installed red blinkers	remarkably complete compendium

Table 2.7: Thirty of the best predicting trigrams from the Brown Corpus.

characteristics of higher-order n-grams in terms of their ability to capture linguistic regularities.

There are well over half a million distinct trigrams in the 1,065,795 word Brown Corpus, and finding terse descriptions of their general characteristics is a considerable challenge. But there are three classes of trigrams of particular interest when it comes to understanding how trigram models respond to both regularity and diversity in large samples of language, and each class exhibits specific characteristics in terms of their ability to capitalise on lexical relationships.

The first interesting class of trigrams is the set made up of those with the highest instantaneous savings over the unigram model. There are about five thousand trigrams in the Brown Corpus that each save as much as twenty bits when encoding the last word, and thirty of these are listed in Table 2.7. One dominant characteristic of these trigrams is that, like the best bigrams, they are comprised entirely of content words. While this observation does not hold for the entire set, it is a dominant feature, and those trigrams that do involve a function word are almost always compound or complex nounphrases. But, again like the best unigram predictors, a more significant characteristic is that these trigrams are all single instance sequences, thus

their savings might more readily be attributed to the problem of using *post hoc* probabilities derived from sparse data (see Section 2.4.1) than to any intrinsically useful linguistic property they might embody.

In comparison, Table 2.8 lists the thirty most *useful* trigrams from the Brown Corpus, where utility is given as the product of the probability of the trigram and its instantaneous savings over the unigram model. Once again, like the bigram model, it appears from this set of trigrams that those which provide the most benefit for lowering complexity estimates for language in general tend to be comprised exclusively of function words.

The third interesting class is that consisting of trigrams with little utility, due to some combined effect of low frequency and poor instantaneous savings. There are over 130,000 trigram types in the Brown Corpus (approximately 15%) which give savings less than five bits over the unigram model, almost 2500 of which actually entail negative savings. Table 2.9 lists thirty of the least useful trigrams, along with their average expected cost in complexity. Just as was the case with poorly performing bigrams, these trigrams are almost entirely ones which combine content words with function words—the vast majority being single semantic terms combined with two grammatical terms.

These results are consistent with the conclusions suggested by the bigram study. First, while n-gram models are able to make substantial gains by capturing semantic relationships between content words, specific instances of these relationships are too infrequent to be useful in general. N-gram models are much better at capturing common grammatical structures manifest as sequences of functions words, provided content words do not get in the way. More to the point, the strengths of the n-gram technique are severely undermined by vast numbers of nearly useless lexical sequences which arise when open-class words are combined with closed-class words. This suggests the possibility that a model that ignored mixed-class n-grams might obtain similar final complexity estimates from a substantially smaller model.

Whether these trends hold as the order of the model increases is very difficult to determine because the problem of data sparseness prevents reliable extrapolation of characteristic behaviours. As the order of the model goes up, the number of possible n-grams increases exponentially, and the amount

	savings	
trigram	instantaneous	expected
as well as	6.83285	0.002519
one of the	3.13477	0.002035
he did not	7.09588	0.001132
at the same	6.59481	0.001052
in new york	11.13862	0.001035
it would be	6.05251	0.000994
in order to	5.01812	0.000933
it is not	4.41524	0.000930
a number of	4.79243	0.000913
mr and mrs	10.96017	0.000815
as a result	8.13856	0.000807
he had been	5.42531	0.000756
i don't know	8.98740	0.000738
on the other	4.66028	0.000722
as long as	7.08824	0.000681
in front of	4.68981	0.000647
out of the	2.28260	0.000615
a couple of	4.72606	0.000549
be able to	5.24272	0.000528
in terms of	4.83164	0.000524
it should be	6.87072	0.000522
more or less	11.18184	0.000520
it was a	2.32467	0.000516
is to be	5.27511	0.000515
most of the	2.90468	0.000513
at the time	4.77769	0.000511
it may be	6.45780	0.000510
had to be	5.11939	0.000499
it has been	6.88634	0.000480
it had been	7.13391	0.000475

Table 2.8: The thirty most useful trigrams from the Brown Corpus.

trigram	cost	trigram	cost
degree of the	-1.714	a letter the	-0.392
a cross of	-1.712	ssociation of the	-0.392
a single and	-1.422	action and the	-0.318
a week the	-1.355	and walked the	-0.318
an hour the	-1.157	carried out the	-0.240
a hundred the	-0.928	a major and	-0.219
a world the	-0.928	a head in	-0.215
children and the	-0.825	a life the	-0.157
deal of the	-0.825	and south the	-0.157
an area the	-0.714	contrary to the	-0.157
an individual the	-0.714	a strong and	-0.114
at night the	-0.714	act of the	-0.114
church and the	-0.714	an american and	-0.078
a car the	-0.655	at present the	-0.070
a state the	-0.593	average of the	-0.070
a war the	-0.593	burden of the	-0.070
bed and the	-0.462	cold and the	-0.070
college and the	-0.462	color and the	-0.070
a child of	-0.449	death and the	-0.070
a group the	-0.392	a rise of	-0.034

Table 2.9: Thirty trigrams with negative savings.

of data required to garner reliable statistics for them (in practice) quickly runs out of reach. For example, just 40% of all distinct unigrams in the Brown Corpus occur only once, but 76% of all bigram types in the Brown Corpus are single instance ones, 92% of the trigrams occur once, 97.5% of the 4-grams (i.e. order-3 contexts) are single instance, and over 99% of five-word contexts occur only once. Thus statements about best and worst n-grams in even modestly high-order models become meaningless because all n-grams become equally good or bad. Certainly the likelihood of finding long word sequences that do not include any closed-class words is greatly diminished and, since this is the lexical distinction of interest within this thesis, further discussion of n-grams is restricted to bigrams and trigrams.

2.4 Model measures

Probability estimates, particularly conditional ones, depend on assumptions about what is or is not relevant to the calculation. All of the statistics cited thus far in the thesis, for example, are based on the assumption that observations made of the Brown Corpus are representative of the true underlying distributions. Such an assumption is quite optimistic, and can easily lead to erroneous conclusions. Imagine if our estimates about the outcome of tossing a fair coin were based on just a few observations. If we happen to see an equal number of heads and tails then a subsequent assertion that they are equiprobable events will be correct, but any other observed distribution would lead to an invalid conclusion. In the worst case, we might not see any heads at all during training and conclude that the probability of such an outcome is zero! Clearly if any event is *possible* then its likelihood cannot be zero.

Given that complexity estimates are relied upon extensively in this thesis when evaluating various language models, we discuss in this section the problems associated with using posterior probabilities. In addition, because the size of a model has a significant effect on estimated distributions, we describe how model compactness and n-gram utilisation can be factored into the analysis.

2.4.1 Over-fitting the data

It is true that bigram and trigram complexity estimates in practical language modeling tasks are typically very difficult to match [18]. This makes them attractive under Occam's principle: that a good theory, or model, is one that provides accurate predictions about the data. However, this has not actually been substantiated in the examples described earlier in this chapter because the entropy estimates have been based on posterior probabilities. Occam's principle is intended to apply to the model's ability to predict future observations. The models outlined in this chapter are in many instances incapable of doing this.

Figure 2.7 plots the number of bigrams gleaned relative to the amount of training corpus processed. While some tapering off is evident, the rate of

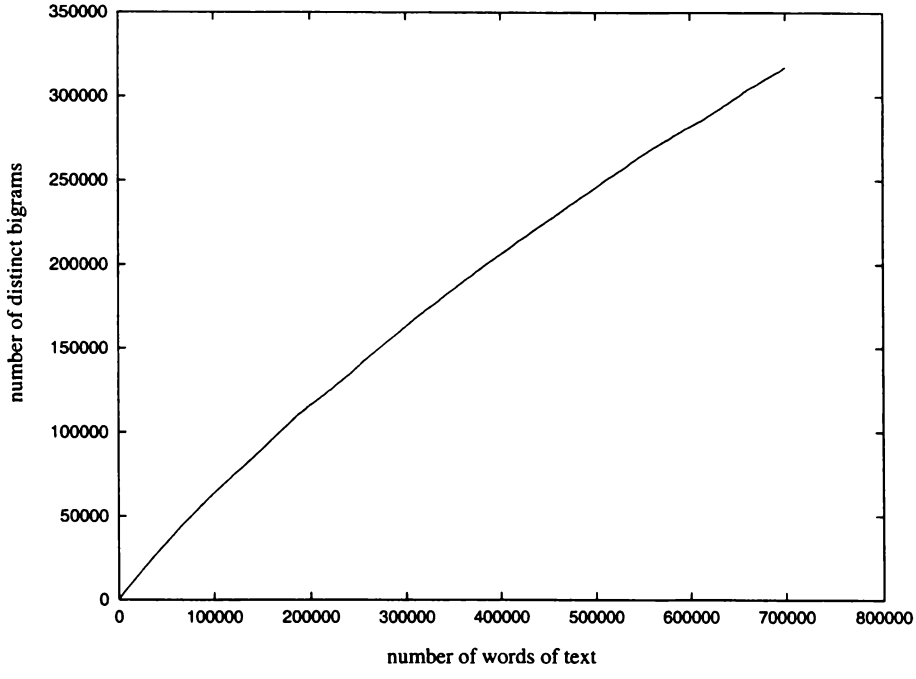


Figure 2.7: Number of bigrams with respect to number of words observed.

growth for the bigram model remains more or less constant. As more and more text is observed the rate of growth diminishes, but the speed with which it diminishes is extraordinarily slow due to the unending occurrence of novel word pairs in the ever growing sample corpus. Assuming that novel n-grams continue to present themselves (at least until sufficient observations have occurred to produce a comprehensive model) there will always be possible future events to which the model is unable to assign a probability.

Unfortunately, there is no rational way to estimate what probability ought to be assigned to an unseen event. Practical language processing tasks deal with this so-called zero-frequency problem by *smoothing*, a heuristic process that prevents any probability from reaching zero while simultaneously attempting to produce a more accurate distribution over observed phenomena. In text compression, for example, the solution is to withhold a small portion of the probability mass from the observed phenomena and assign it to an escape procedure for dealing with novel events. How large or small a probability one should assign to unseen events can be adjusted based upon expectations about how often such events might take place and this has been

studied extensively [8, 22]. There is, however, some justification for just ignoring the zero-frequency problem in this study.

2.4.2 Bootstrap assumptions

One might be tempted to argue that (unlike practical language problems) language modeling, as an instance of grammatical inference, can ignore unseen n-grams all together. Grammatical inference is generally viewed as a bootstrapping problem such that the learning mechanism must simultaneously acquire a characterisation of lexical relationships and assign probabilities to them [43]. If a particular word sequence has not been observed (under the n-gram paradigm) then the learner has no motivation to include it in the model and thus no probability need be assigned to it. That is, there is no such thing as an unseen sequence as far as the learner is concerned. When a novel event does occur, the learner simply adds it to the model and assigns some appropriate non-zero probability to it based on its relative frequency.

There are other reasons why it is not unthinkable to ignore the zero-frequency problem. For a sufficiently high-order model, there may be some lexical combinations which, perhaps due to some fundamental linguistic constraint, can never occur, thus assigning probabilities to them would violate the principle of a stochastic generative model: that the sum of all sentence probabilities is one. Further, some combinations may indeed occur in speech, but not because they are part of the language *per se*. For example, a stuttering speaker may start a sentence with "the the", creating a situation where the learner must decide whether to include this as a viable bigram or dismiss it out of hand as a production error that does not reflect an underlying linguistic principle that must be learned. This is somewhat analogous to Quine's parable about the person who does not believe in UFO's but is then confronted with what appears to be an extra-terrestrial alien. The observer may change his model of belief to admit the existence of such aliens, or dismiss the experience out of hand and carry on with his original mental precepts.

2.4.3 Assumption of generality

There are equally cogent reasons why ignoring the zero-frequency problem is untenable. First, if the model does not have to account for unseen events then there is presumably nothing to stop it from assuming a single, very long context when estimating the conditional probability for a word. That is, after observing k words, the model maximises its estimate of input complexity by assuming a $(k-1)$-order model and thereby maintains a perpetual capacity for perfect prediction from a linearly growing model.

Second, without the ability to accommodate unseen n-grams the model is not adequately generative and thus must be implicitly overlooking some kind of generalisation that could lead to better predictions. More plainly, without a comprehensive set of n-grams there are likely to be some word sequences that form part of the language but which cannot be produced by the model, thus the model has failed to capture some fundamental abstraction that accounts for language regularities and, presumably, could therefore be exploited for improved prediction. One of Pinker's students, for example, presented him with the most unusual sentence "buffalo buffalo buffalo buffalo buffalo buffalo buffalo buffalo buffalo buffalo" along with an adequate explanation as to why it is wellformed [85]. The explanation relates to acceptable complex relative clause embeddings and (obviously) polysemy and, if nothing else, shows that n-gram models miss a great deal of linguistic regularity in their superficial treatment of word combinatorics.[1]

2.4.4 Maximum likelihood estimation

Despite these objections, there is still sufficient reason why it is acceptable in this study to overlook the zero-frequency problem when calculating complexity estimates from n-gram models. The goal of this thesis is not to propose a solution to the language modeling problem, but to observe how the assumption of independent sequential dependencies for function words and content

[1]The *buffalo* sentence parses because there is a city in the U.S. called Buffalo, and buffalo that come from there might be called Buffalo buffalo. Further, there is a verb "buffalo" which means to intimidate. Thus the sentence might be rewritten as "Buffalo that come from Buffalo that are known to buffalo other buffalo that come from Buffalo are also known to buffalo other buffalo from Buffalo that are themselves known to buffalo some buffalo from Buffalo."

words affects a model's ability to predict the next word. More generally, we want to compare models to see how changes to the underlying formalism influence their performance, assuming a stationary source. It is desirable to make such comparisons under the most favourable conditions possible—specifically, using parameter values that give the highest probability to the training data without wasting any of the probability mass. The estimate that does this is the *maximum likelihood estimate* [76], a likelihood function that assigns probabilities based solely on relative frequency.

2.4.5 Minimum description length

One danger from the maximum likelihood assumption is that it entails an implicit bias in favour of models that use very long contexts to predict the next word. High-order models end up giving extremely good entropy estimates because so many of their n-grams predict the next word with near or absolute certainty. The penalty for this is overly large models whose deep contexts are observed too infrequently to capture generally useful linguistic patterns.

Solomonoff's theory of inductive inference views learning as the problem of "finding a shorter description of the observed data" [107], thus it is desirable to factor the size and complexity of a model into the measure of its effectiveness. A general formula for estimating the quality of a model in terms of both its predictive capacity and its size is given by the *minimum description length principle* [89], which states that the best theory to explain a set of data is one that minimizes the sum of 1) the length, in bits, of the description of the theory, and 2) the length, in bits, of the data encoded with respect to the theory [51].

For language modeling, the number of bits needed to encode a language sample given a model is the sum of the entropy estimates derived from lexical probabilities. Measuring model size, however, is not quite so straightforward, and depends a lot on the chosen representation. Certainly the number of n-grams is of interest, but there are more efficient ways of encoding a set of n-grams than simply enumerating them. If we represent bigrams as a sorted list, for example, we need only itemize the first term whenever it is not the same as the first term of the preceding bigram. If the bigrams are not sorted

then a list of the same format would likely have very many more terms and might require many more bits to encode, thus the former method seems preferable. But it could be that a sorted list of bigrams with their terms reversed would require even fewer bits to encode.

Other more principled methods have been proposed. The Russian mathematician, A. N. Kolmogorov, suggested that the complexity of a model could be measured in terms of the number of bits required to express it as a computer program [72], and Muggleton et al. [82] measured it in terms of the number of bits needed to disambiguate choice points in a corresponding proof tree. But, just as it is not possible to assign correct probabilities to unseen events, neither is it possible to compute the exact complexity of a model.

2.4.6 Performance metrics

In this thesis, four simple measures are used for making comments about the relative merits of a model. First is its ability to assign low complexity estimates to language samples. Second, its absolute size in terms of the number of n-grams actually observed in the training sample. Third, its compactness with respect to a comprehensive model of the same form. And fourth, the spread of utilisation over the observed n-grams—giving particular consideration to the number of single instance n-grams. In addition to these four physical metrics, informal linguistic intuitions are applied in an analysis of how well models capture *bona fide* attributes of language structure.

2.5 Discussion

N-gram models characterise lexical relationships directly by modifying estimates of the distribution over the vocabulary based upon the context of the words given in the prior context. But n-gram models are undermined in situations where the context does not contain the best evidence for forming an accurate prediction. It is not always the case that immediately preceding words offer the best clue as to what word will appear next in an expression. For example, in the sentence

The result is either overheating of the manifold or failure of the valve.

we observe that the occurrence of the word "or" is rather more related to the prior occurrence of "either" than it is to any other preceding word. To some extent, this relationship might be exploited in a finite context model by increasing the length of the context so as to include "either", but this is an unsatisfactory solution as it leads to an exceedingly large model. Moreover, it would not help anyway when there are, say, six or seven words between "either" and "or".

There is no doubt that n-gram models provide good complexity estimates for language, and they are straightforward to construct and easily applied to many practical language processing problems. But, as far as their ability to exploit real linguistic regularity is concerned, they are severely limited in what they can achieve. Aside from their inherent inability to capture certain kinds of recursive structure, such as centre-embedding, they overlook many aspects of lexical dependency which can in principle be expressed within a regular grammar formalism—specifically, categorial dependencies and some forms of agreement.

This chapter shows that n-grams are very good at taking advantage of strong lexical relationships between adjacent content words—relationships that presumably arise because of an underlying semantic association. Unfortunately, such n-grams are also extremely rare, making them largely unhelpful for delivering low complexity estimates for language in general. This situation could be moderated if the incidence of such adjacencies could be increased. The next chapter outlines an alternative lexical model that achieves the same effect by exploiting joint probabilities for non-adjacent words.

N-grams are also good at capitalising on frequent patterns of functional terms—patterns that arise in response to syntactic requirements given by the language. One limitation to the model's ability to take better advantage of these high-frequency terms is that intervening content words create tremendously diverse n-grams, and this obscures the underlying regularity: the categorial relationship between function words and content words. If content words could be replaced with an appropriate category symbol, then the diversity of these n-grams would diminish and the fundamental relationship would become more salient. This proposal is explored in Chapter 4.

Chapter 3

Lexical Attraction Models

The n-gram studies outlined in the previous chapter indicate that uninterrupted patterns of grammatical terms are well utilised under such a paradigm, but contiguous sequences of content words are not. While strong semantic relationships between content words have potential for translation into low complexity estimates, their relative infrequency undermines their general utility. Furthermore, though local semantics tend to give rise to close proximity for related content words, the adjacency required by n-gram models is seldom satisfied. It is hypothesised that both of these effects might be moderated if intervening function words could be ignored in the conditioning contexts for content words, and that substantial improvements in probability estimates from a more compact model would result.

This chapter describes the lexical attraction model—a technique that allows joint probabilities for nonadjacent terms to be incorporated into the predictive mechanism and thus exploits long distance lexical dependencies in estimates of language complexity. An overview of precursive grammatical formalisms and statistical effects in language that suggest lexical attraction is given, followed by descriptions of the algorithms required to identify lexical links.

Results from the application of lexical attraction to practical language processing tasks show that while such models are good at identifying long distance dependencies, the overhead required to specify the dependencies negates the potential savings they afford. Observations about the characteristic behaviour of the lexical attraction model, however, indicate that its

fundamental strengths can be preserved within the much simpler formulation of n-grams, eliminating the need to specify lexical links altogether. This idea forms the basis for the super-adjacency models outlined in Chapters 5 and 6.

3.1 Lexicalist grammars

The traditional structuralist view of syntax, begun in the 1950's, regards grammar as a kind of mechanical specification for the proper construction of sentences. The predominant formalisms define a production relation characterising the manner by which words can be combined to produce well-formed expressions. The production relation describes syntactic derivations obtainable through substitution rules expressing increasingly refined subcategorization properties for lexical heads—nouns, verbs, and so forth—and their complement structures [26].

In the late 1970's, some linguists were developing misgivings about purely structuralist accounts because of their inability to support strong statements about universal grammar—something even generative-transformational grammars could not satisfactorily accomplish [78]. Johnson and Postal [61] responded by developing an approach to syntax in which rules could be formulated directly in terms of grammatical relations between terms (such as between subject and object) instead of relying on purely structural associations. While this so-called relational grammar gained many adherents, it lost momentum for two reasons. First, it lacked publication of a clear statement about many of its details, instead finding its supporters chiefly by word of mouth. Second, the development of lexical-functional grammar (LFG) by Bresnan [14] superseded relational grammar with an extreme lexicalist account of language that reduced structural rules to a subordinate role.

3.1.1 Limitations for structural models

Significant weaknesses for n-gram models arise because of their fundamental restriction to the class of regular languages. This means they are unable to exploit some kinds of recursive structure (such as center-embedding) or long distance dependencies (such as agreement constraints) which could be useful

when making predictions about language. The first of these can be overcome by adopting a stronger structural formalism, such as a context-free grammar, which treats sentence structures as a hierarchy of embedded substructures, or constituents. But context-free grammars are unable to capture long distance dependencies without facing an exponential increase in size. For example, consider the following simple context-free grammar for a small subset of English:

$\mathbf{G_1}$

S	→	NP	VP
NP	→	Det	N
VP	→	V	NP
Det	→	a	
Det	→	the	
N	→	dog	
N	→	cat	
V	→	chases	
V	→	likes	

The language of this grammar has 32 sentences, such as "the dog chases a cat" and "the cat likes the dog". The language can be extended quite simply by increasing the terminal vocabulary—for example, by adding the new noun rule

$$\text{N} \rightarrow \text{mouse}$$

Addition of this rule increases the number of nounphrases from 4 to 6, more than doubling the size of the language as a consequence. So it is that rules may be continually added to introduce new nouns, verbs and determiners into the vocabulary, with each new rule yielding a multiplicative increase in the size of the language. However, when a terminal is added that embodies a new inflectional form, the grammar itself may have to be modified to ensure agreement, and this can give rise to a multiplicative increase in the size of the grammar for the sake of adding one new word. For example, to add the plural noun "dogs" to the language of $\mathbf{G_1}$ we must modify the grammar to preserve number agreement between the determiner and noun, and between the noun and main verb, as with

G_2

S	→	NPs	VPs	Det	→	a
S	→	NPp	VPp	Det	→	the
NPs	→	Det	Ns	Ns	→	dog
NPp	→	the	Np	Ns	→	cat
NP	→	NPs		Np	→	dogs
NP	→	NPp		Vs	→	chases
VPs	→	Vs	NP	Vs	→	likes
VPp	→	Vp	NP	Vp	→	chase
				Vp	→	like

In this instance the grammar nearly doubles in size just to accommodate the new plural noun—though admittedly ancillary verb forms have had to be introduced as well. Once the new inflectional form has been allowed, of course, new terms similarly inflected can be added with impunity. But the number of possible agreement constraints can be quite large—encompassing all noun and verb subcategorizations, case, and defective verb requirements, to name but a few—and if semantic agreement is also supported then the size of the rule set can become exceedingly large. Bresnan summarises the situation as follows:

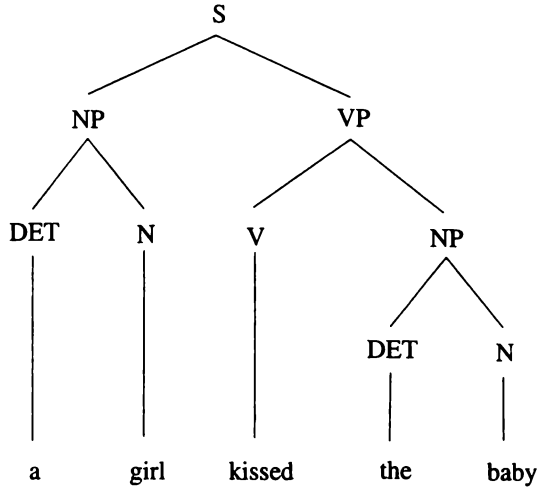
> The fundamental problem for a theory of syntax is to characterise the mapping between semantic predicate-argument relationships and surface word and phrase configurations by which they are expressed. This mapping is sufficiently complex that it cannot be characterised in a simple, unadorned phrase structure formalism: a single set of predicate-argument relations can be realised in many different phrase structures (e.g. active and passive constructions), and a single phrase structure can express several different semantic relations, as in cases of ambiguity.
> ([15], page 174)

Bresnan's lexical constraint grammar avoids this problem by transferring the responsibility of agreement conditions (and much more) to the words themselves.

3.1.2 Lexical-functional grammar

LFG represents a marked departure from structuralist grammars in that the tasks of enforcing proper word order and satisfying agreement constraints are

c-structure



f-structure

$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{SPEC} & \text{A} \\ \text{NUM} & \text{SG} \\ \text{PRED} & \text{'GIRL'} \end{bmatrix} \\ \text{TENSE} & \text{PAST} \\ \text{PRED} & \text{'KISS}\langle(\text{SUBJ})(\text{OBJ})\rangle\text{'} \\ \text{OBJ} & \begin{bmatrix} \text{SPEC} & \text{THE} \\ \text{NUM} & \text{SG} \\ \text{PRED} & \text{'BABY'} \end{bmatrix} \end{bmatrix}$$

Figure 3.1: A sample lexical predicate argument structure.

almost entirely given to the lexicon. Mapping predicate argument structures to surface forms is lexically encoded using universal grammatical functions, with phrase structure computations playing a superficial role [15]. One direct consequence of this so-called *principle of direct syntactic encoding* is that each inflection for a particular root word form must have its own lexical entry because they imply different grammatical relations. But by moving details of grammatical function to the lexicon, syntactic characterisations are greatly simplified.

LFG assigns two levels of syntactic description to every sentence. First, a constituent structure (or *c-structure*) is used to indicate the superficial arrangement of words and phrases within a conventional phrase structure

tree. Second, surface grammatical functions are represented using a functional structure (or *f-structure*) which provides a precise characterisation of syntactic relations, such as subject, direct object, complement, adjunct, and so forth. Whereas c-structures are defined in terms of syntactic categories, capturing dominance and precedence relationships, f-structures are defined using grammatical function labels, semantic forms and feature values (and subsidiary f-structures).

Figure 3.1 provides an example of the LFG syntactic description for the sentence "A girl kissed the baby". The c-structure is given by a context-free grammar characterising all possible surface structures for the language (without transformations). However, unlike conventional rewriting systems, righthand sides of grammatical rules give a functional specification for an expression, not an explicit derivation process for combining words into sentences.

Functional specifications consist of *statement schemata* which are, roughly speaking, precedence rules annotated with *metavariables* for capturing f-structure information. For example, consider the following schemata for the statement in Figure 3.1 which express the standard phrase structure constituent order: a sentence consists of a nounphrase followed by a verbphrase.

$$\begin{array}{lcll} \text{S} & \rightarrow & \text{NP} & \text{VP} \\ & & (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \\ \\ \text{NP} & \rightarrow & \text{DET} & \text{N} \\ \\ \text{VP} & \rightarrow & \text{V} & \text{NP} \\ & & & (\uparrow \text{OBJ} = \downarrow) \end{array}$$

The metavariable assignment $(\uparrow \text{SUBJ}) = \downarrow$ in the S schema indicates that the subject f-structure comes from the NP immediately dominated by S. More plainly, the $\downarrow$ metavariable refers to the node's own f-structure (i.e. that of the subject NP), and the $\uparrow$ says that the same f-structure is a subsidiary within the parent node's SUBJ f-structure (i.e. for S itself). Put another way, arrows that point to each other across one line in the corresponding parse tree are instantiated with the same f-structure value.

The metavariables become instantiated with *actual variables* when a rule is applied. The actual variables capture specific syntactic and semantic fea-

tures, and ultimately become grounded by terminal symbol schemata in the lexical entries. Lexical schemata include part-of-speech information and inflectional features for each word of the language. For example, the lexical entries for the sentence of Figure 3.1 are

a:	DET,	(↑ SPEC) = A
		(↑ NUM) = SG
the:	DET,	(↑ SPEC) = THE
girl:	N,	(↑ NUM) = SG
		(↑ PRED) = 'GIRL'
baby:	N,	(↑ NUM) = SG
		(↑ PRED) = 'BABY'
kissed:	V,	(↑ TENSE) = PAST
		(↑ PRED) = 'KISS⟨(SUBJ)(OBJ)⟩'

Note that lexical schemata are of the same form as the grammatical schemata of c-structure rules allowing uniform treatment during instantiation. All metavariables for words are of course ↑ ones because lexical items are always nondominating (i.e. terminal symbols).

Variable instantiation occurs in three phases. First, grammatical schemata are attached to appropriate c-structure nodes as given by the rules, and the lexical schema for each word is attached to its immediately dominating preterminal node. Second, initial f-structure forms are created for each metavariable and instantiated with any details specified by the schemata. Finally, each ↓-variable is instantiated within the f-structure at that node by merging with the ↑-variables of all the nodes it immediately dominates. Provided there is no contradiction in merging the details of f-structures at adjacent levels, the f-structure for the root node gives a complete functional description for the sentence. That is, a successful merge requires that all immediately dominated nodes *agree* as to the specific value of each syntactic feature within the f-structure of their common parent.

Merging of f-structure details is recognizably the process of *unification*, which was just making its way into computational linguistics at the time LFG was first introduced. Colmerauer [32] had only recently shown how to support unification within logic programs, and Kay [63] had demonstrated how it

could be used to maintain and pass grammatical information during a parse. The feature schemata of LFG proved a perfect match to the parameter vectors of logic programming, demonstrating the viability of unification-based grammatical formalisms within a computational framework [78].

3.1.3 Limitations of feature-values

One of the most important aspects of LFG is its potential to eliminate the exponential growth inherent to conventional phrase structure grammars when attempting to address the problem of agreement. Instead of incorporating large numbers of special purpose rules to deal with each form of inflection, as demonstrated earlier in this chapter by grammar $\mathbf{G}_2$, a substantially smaller set of general purpose rules can be annotated with feature parameters. Thus, for example, $\mathbf{G}_2$ can be expressed more tersely with the following unification grammar:

$\mathbf{G}_3$

S	$\rightarrow$	NP(X)	VP(X)
NP(X)	$\rightarrow$	Det(X)	N(X)
VP(X)	$\rightarrow$	V(X)	NP(Y)
Det(sing)	$\rightarrow$	a	
Det(_)	$\rightarrow$	the	
N(sing)	$\rightarrow$	dog	
N(sing)	$\rightarrow$	cat	
N(plur)	$\rightarrow$	dogs	
V(sing)	$\rightarrow$	chases	
V(sing)	$\rightarrow$	chases	
V(plur)	$\rightarrow$	chase	
V(plur)	$\rightarrow$	like	

The grammatical structure of a sentence and its constituents is characterised through more general rules by transferring agreement details to feature parameters. Actual feature values are given by lexical rules, and percolate up through the parse tree by way of unification. The feature parameters thus permit conveyance of dependency constraints between distant constituents, such as the required number agreement between the subject nounphrase and main verb in $\mathbf{G}_3$.

The array of syntactic and semantic agreement constraints can grow without limit by extension to the parameter vector at any node. Thus if it is

desirable, say, to prevent "colourless green ideas" from "sleeping furiously" one could introduce a feature which validates the capacity for "sleep" for just those nouns to which it seems appropriate. Such a feature may extend to the concept of, say, "waking" as well, allowing more general application of a particular feature value. However, the ability to increase the number of sustainable agreement constraints highlights two important shortcomings of feature-value formalisms. First, the savings in grammar size are somewhat misleading in that, while the size of the syntactic description for constituent structure is indeed reduced, it comes at an increased cost in the size of the lexicon. Second, and more importantly, the decreased complexity with respect to grammar size is simply transposed into increased complexity in the execution of the unification process. The grammar may be smaller but its application in generative procedures is substantially more complex.

3.1.4 Link grammar

The idea of feature agreement is of interest for language modeling in that the existence of lexically-driven constraints must influence probability estimates for the occurrence of one word given another. That is, if the feature matrix of one word imposes agreement restrictions on others then this can be exploited for predictive purposes. But to establish lexical relations through a hierarchical structure, as LFG does, is an unnecessary overhead. Notions about constituents and phrasal categories can be ignored if lexical dependencies can be established directly.

Sleator and Temperley [97] devised the *link grammar* as an unconventional formalism for capturing agreement constraints as direct links between words. A link grammar connects pairs of words in a sentence using undirected labelled arcs. A valid parse exists if the grammar defines at least one linkage for the entire sentence which results in a connected planar graph. For example, Figure 3.2 shows a link parse for a sample sentence of English.

In Sleator and Temperley's original system, labelled links correspond to grammatical relations between words. The labels shown in Figure 3.2, for example, are interpreted as follows: `S` connects a noun to its verb, `EV` connects a verb to its modifying prepositional phrase, `O` connects a verb to its object, `D` connects a determiner to its noun, and `J` connects a preposition to its object.

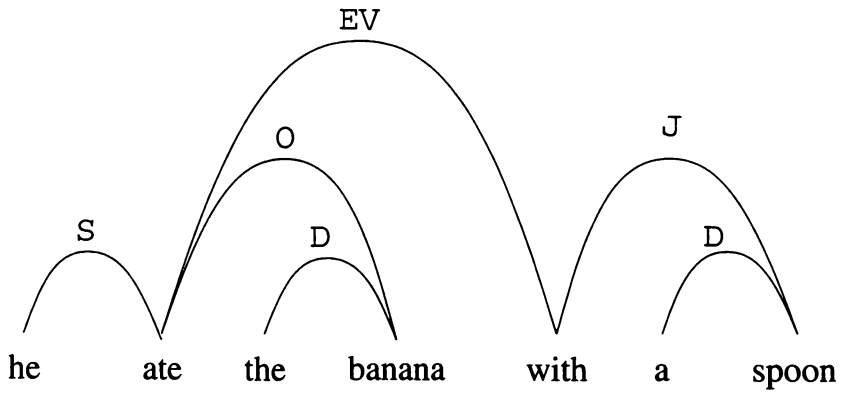


Figure 3.2: A sample link grammar parse.

Their vocabulary of labels captures 107 different grammatical relations—roughly the same size as a conventional set of part-of-speech labels that might be found in a standard phrase structure grammar.

Matching rules are specified in the dictionary as a sequence of connector specifications which define linking requirements for individual words. The connectors specify kinds of links a word may be involved in, and each connector is followed by either + or - to indicate whether the word attaches respectively at the left or right end of the proposed link. A match occurs when two words can be found which satisfy the connector at opposite ends of the link.

To that end, each entry in the dictionary consists of a set of words associated with a common matching rule defining how such words may be linked to others. A rule is expressed in a disjunctive form wherein connectors are combined using the binary associative operators `&` and `or`. All lefthand connectors in the rule must be satisfied, along with any righthand connectors which follow. For example, the following entries define link requirments for a handful of nounphrases.

```
a:                Ds+
big black ugly:   A+ or (AI- & {@EV+})
dog cat stick:    {@A-} & Ds- & {@M+ or (C+ & Bs+)}
                  & (J- or O- or ({C- or CL-} & Ss+) or SIs-)
```

The partial linkage these rules give for the nounphrase "a big black dog" is shown in Figure 3.3. The `Ds+` requirement for "a" indicates it must connect

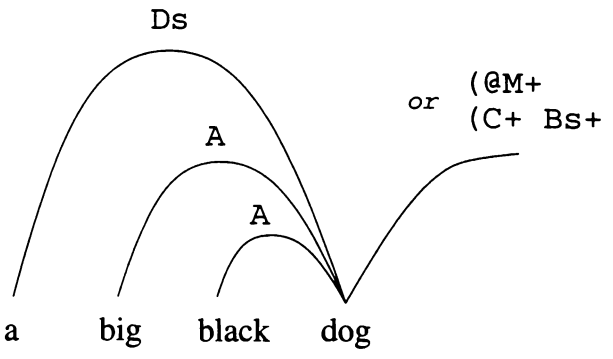


Figure 3.3: A partial linkage for a simple nounphrase.

rightward with a Ds-. This is satisfied by the connector matching rule for the noun "dog", but the noun's rule also allows zero or more leftward links with a A+, as specified by its connector @A-, where the @ means *any number of*. These links are achieved with the two adjectives "big" and "black". The noun's matching rule further indicates that it has an outstanding linkage requirement of type @M+ or C+ & Bs+, which must extend rightward. The remainder of the noun matching rule is ignored because any other interpretation would require different leftward links than are possible (such as a prior J+). (Note that lowercase letters of connector labels are conventionally subscripts for allowing more general link requirements, such that Ds+ can link with Ds- or just D-, but not with Dp-.)

Unlike phrase structure formalisms, including LFG, words that are associated syntactically and/or semantically within a sentence are directly linked to each other after parsing, instead of having their relationships established by a hierarchical constituent structure. This high degree of lexicality suggests that a probabilistic form of the grammar might perform well for language modeling.

3.1.5 Stochastic link grammar

The basic operation of phrase structure grammars is the application of rewrite rules. Parsing with phrase structure systems can be made more efficient when rules are annotated with probabilities—as with stochastic context-free grammars [22]—by allowing more likely productions to be explored first when

seeking a derivation. Analogously, the basic operation of link grammars is linking. So it is that the process of finding a linkage (or the best linkage) can be improved by choosing links according to a probability distribution.

A link depends upon two things: the direction of the link, and the choice of word in that direction to link to. These in turn depend upon choosing an appropriate disjunct for the words of the sentence. Given that links are a symmetric binary relation, parsing sentences left-to-right eliminates the direction problem by reducing the search to rightward links only. Thus assigning a probability to a linkage involves assigning a probability to a disjunct and to the resolutions for its rightward links. This is similar to bottom-up parsing with probabilistic context-free grammars, and Lafferty *et al.* [69] outline a dynamic programming algorithm, analogous to the inside-outside algorithm [59] for PCFGs, which calculates maximum-likelihood estimates from probabilistic link grammars.

What is important about probabilistic link grammars is not so much the details of their parsing algorithms, but the observation that they are a constrained context-free formalism. This means they subsume probabilistic context-free grammars and, consequently, n-gram models; but are stronger in that they are able to capture long distance dependencies. However, while they are released from the demands of the unification process required by LFG, link grammars must still satisfy predefined grammatical relations, and this incurs heavy penalties in the form of parsing complexity and the need for a substantial amount of prior grammatical knowledge. Given that language modeling seeks to exploit only the predictive aspects of language structure, it would be useful to find a way to preserve syntactic associations between words without the baggage of subcategorization features or abstract grammatical relations.

3.2 Stochastic lexical relations

LFG and link grammar each couch lexical relationships in terms of higher level abstractions—syntactic associations which require satisfaction of either feature agreement or connector constraints respectively. But language modeling need not be concerned with labeling a lexical relationship. It is sufficient

to establish that some relationship exists, and even then only if such knowledge increases a model's ability to predict the next word in an expression. Further, it is not even necessary to establish whether the relationship is linguistically genuine (however one might wish to define such a notion), though one might suspect that actual relationships manifest stronger statistical dependencies.

The challenge for the language modeler is to find a way to capitalise on the statistics of direct lexical associations. Finite context models can avail themselves of relationships between adjacent words, but once the distance between two related words exceeds the size of the context, conventional n-gram techniques begin to break down. Moreover, a context may include words which do not participate in the relevant relationship and thus they create unnecessary overhead by increasing the size of the model dramatically without providing a reciprocal improvement in its ability to predict words. To liberate the statistical language model from the constraints of fixed preceding context it is necessary to allow the search for lexical dependencies to extend as far back from the current word as is practical and useful, and for it to be able to drop from the conditioning context any words which do not contribute to an improved joint probability. More plainly, the objective is to develop a systematic yet tractable method for compilation and search of relevant cooccurrence statistics for non-adjacent terms.

3.2.1 Semantic latency

Beeferman *et al.* [7] note that language, when viewed as a stochastic process, is highly nonstationary. As a discourse unfolds, the topic at hand changes and the vocabulary along with it. The local distribution over the vocabulary is thus distorted over time in favour of words with meaning related to the current semantics.

One particular nonstationary property observable in local distributions is that occurrence of a word tends to increase its likelihood of appearing again quite soon—an effect called *locality of reference*, or more specifically *semantic latency* [38]. As an example, the word "madrigal" appears five times in the million-word Brown Corpus, and the last time it is used is only 193 words after its first appearance. While the unigram probability for "madrigal" is

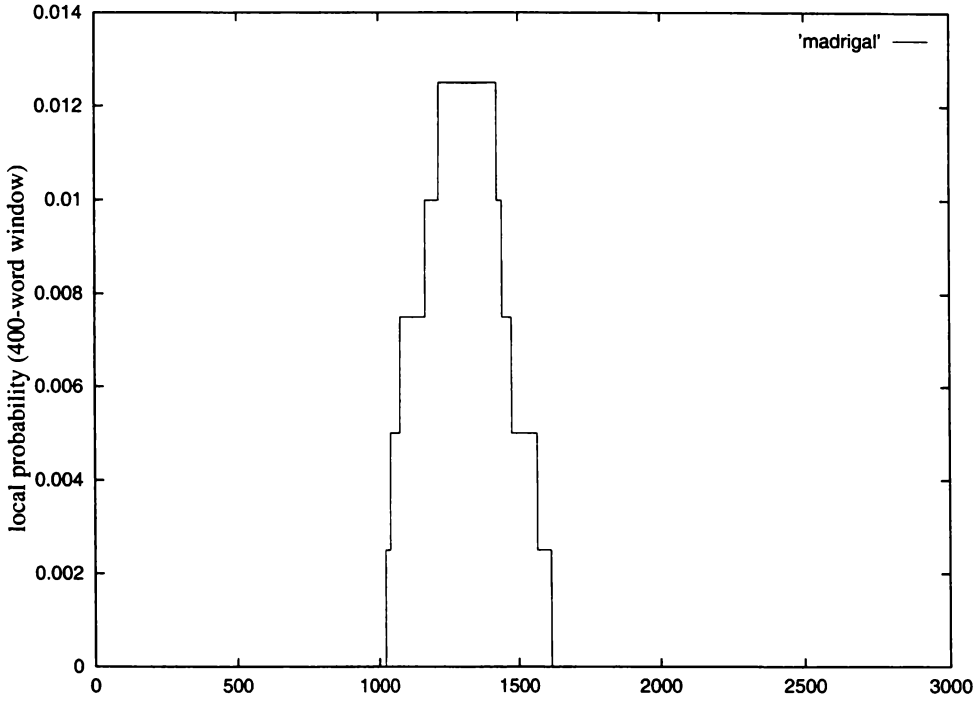


Figure 3.4: Semantic latency effect for "madrigal".

exceedingly small (i.e. $5/1,065,795 \approx 0.0000047$) for the corpus as a whole, a model that estimates probabilities based upon local statistics can increase the likelihood of "madrigal" for the time the text is concerned with them.

The graph in Figure 3.4 illustrates the semantic latency effect for "madrigal". It shows the changing local probability for "madrigal" within a 2500 word segment of the Brown Corpus roughly centered around the 200 word window in which the word appears, where the local probability is calculated as the number of times "madrigal" occurs in the last 400 words.

As the name suggests, semantic latency is a statistical phenomenon of content words only. Given that function words serve syntactic purposes more or less independently from discourse semantics, their local probabilities tend not to vary too far from their global probabilities. Figure 3.5 illustrates this for five of the more common grammatical terms of the Brown Corpus within the same 2500 word segment used in Figure 3.4. Once again, the local probabilities are calculated based upon a 400-word window.

One technique that attempts to take advantage of semantic latency is

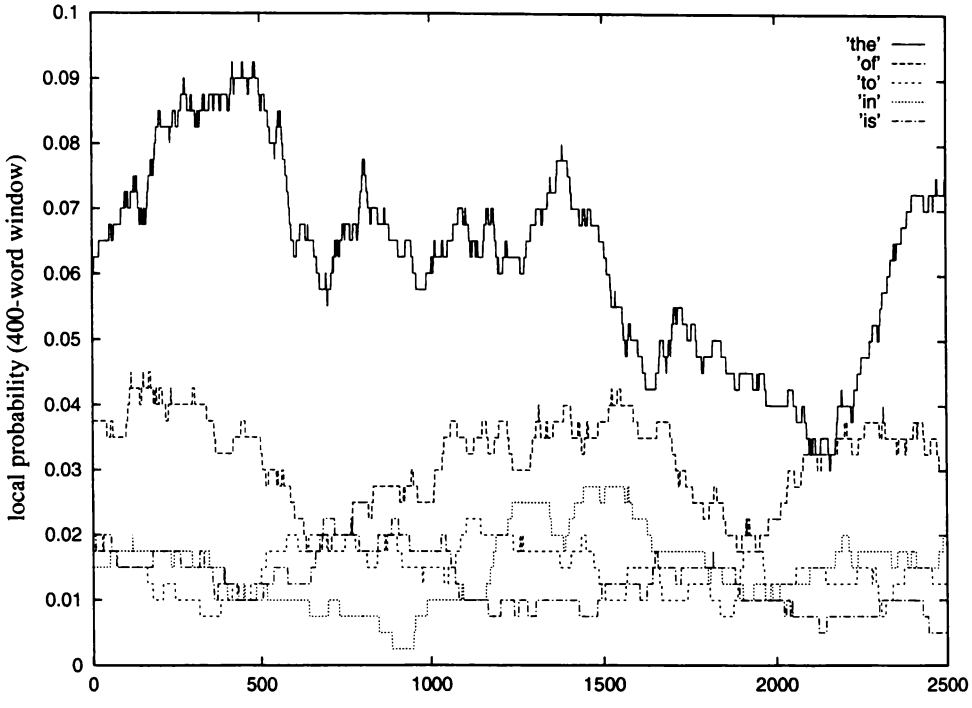


Figure 3.5: Semantic latency of five function words in the Brown Corpus.

move-to-front (MTF) dictionary coding [92]. A MTF encoder maintains a list of the observed vocabulary sorted according to the most recently used term. Each word is encoded explicitly the first time it is encountered but thereafter with its corresponding index in the list. When instances of a particular word appear in unusually close proximity, that word remains near the top of the list, allowing its index to be encoded with just a few bits. This is tantamount to increasing its probability in response to semantic latency. As the topic of discourse drifts on to new themes, the word descends further and further down the list, requiring more bits to encode its index and thus effectively decreasing its probability.

Bentley *et al.* [10] argue that some sequences can be encoded very efficiently under MTF. For example, when a sequence is sorted then all indices are simply one. For typical language samples, however, MTF provides no better overall complexity estimates than does a unigram model. This is not surprising in light of a proof given by Bentley *et al.* which shows that, for a discrete memoryless source, the expected number of bits required to encode

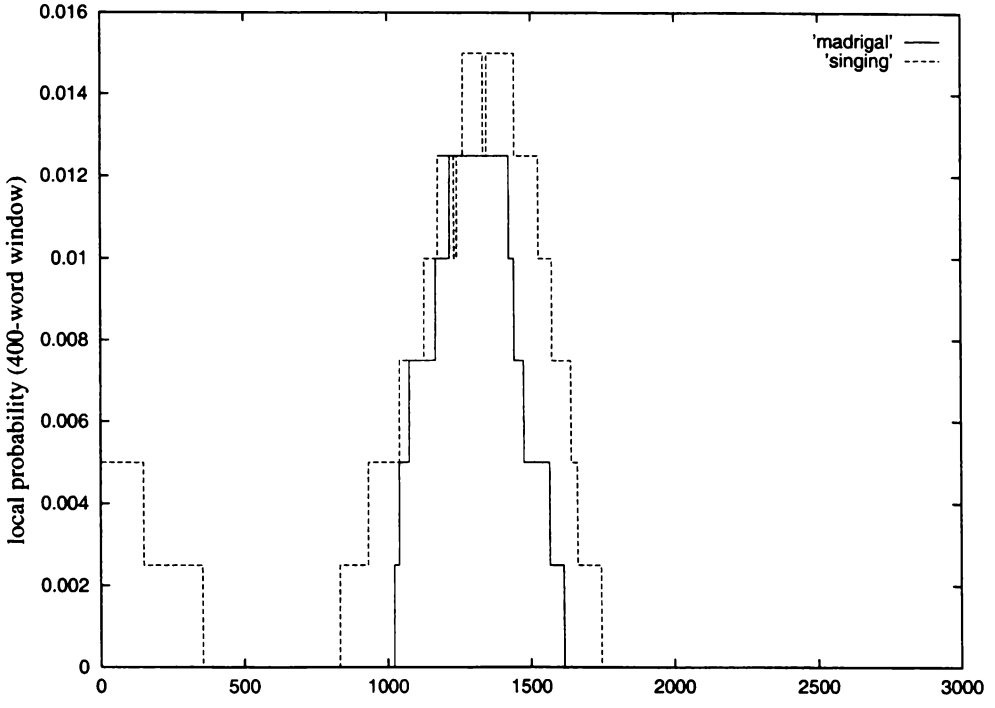


Figure 3.6: Semantic latency of "madrigal" and "singing".

a word using MTF approaches the entropy of the source.

3.2.2 Topic latency

Semantic latency is a highly specialised form of a more general statistical effect—that being an increased joint probability for two terms involved in a semantic dependency. The increased local probabilities for a semantic term are not so much attributable to recent prior occurrence of that term as they are to the more general local discourse semantics in which that word participates. For example, there is an obvious relationship between the words "madrigal" and "singing". As long as the topic at hand pertains to madrigals, the local probability of "madrigal" will likely increase, but the local probability of "singing" might be expected to increase as well for more or less the same reason. That is, local discourse semantics can lead to increased probabilities for a set of related terms such that occurrence of any one term in that set may provide sufficient context to signal increases for all.

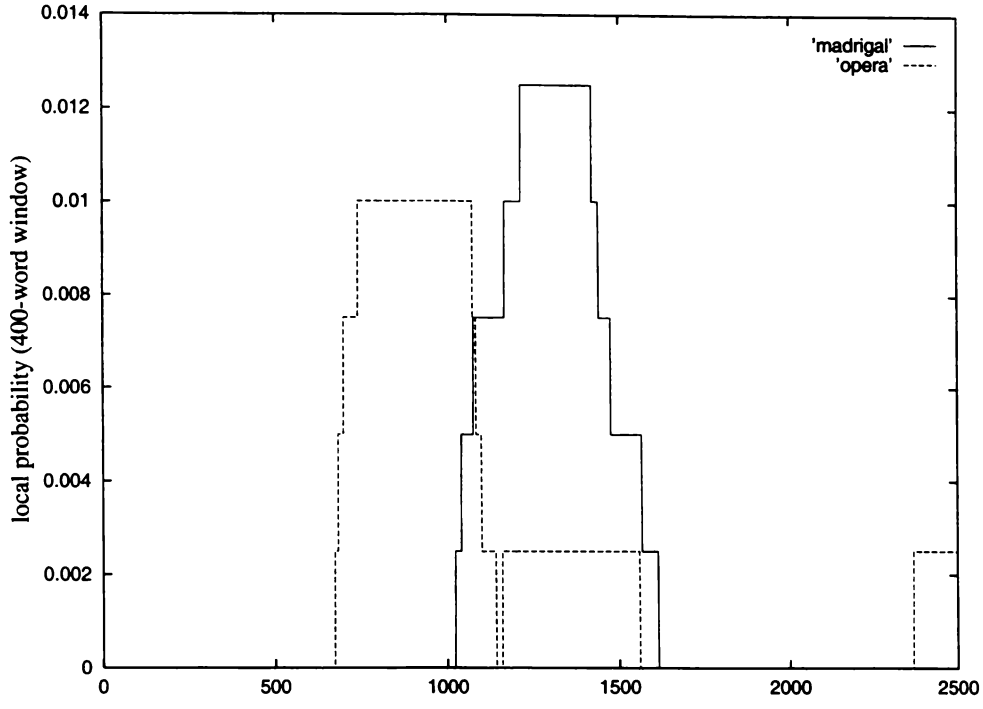


Figure 3.7: Semantic latency of "madrigal" and "opera".

A graph of the local probabilities for both "madrigal" and "singing" is shown in Figure 3.6 using the same segment of the Brown Corpus as before. The graph shows a coincidental semantic latency for the two words, and this suggests a common triggering condition that might intuitively be attributed to the local topic. In comparison, Figure 3.7 depicts the semantic latency for "madrigal" and "opera" for the same text segment. While the close proximity of the increased local probabilities for the two terms suggests some degree of semantic relationship, adjacency of their latency regions instead of overlap seems to imply a sequential treatment of distinct topics. This perhaps further suggests that the latency of both terms might be related to an even broader subject of discourse semantics. While the semantic latency of "singing" partly bridges the two regions, the graph in Figure 3.8 gives evidence of an even more general topic. It plots local probabilities (using a 600-word window) for "madrigal" and "opera" within a broader (6000-word) segment of the Brown Corpus, along with the encompassing semantic latency of "music".

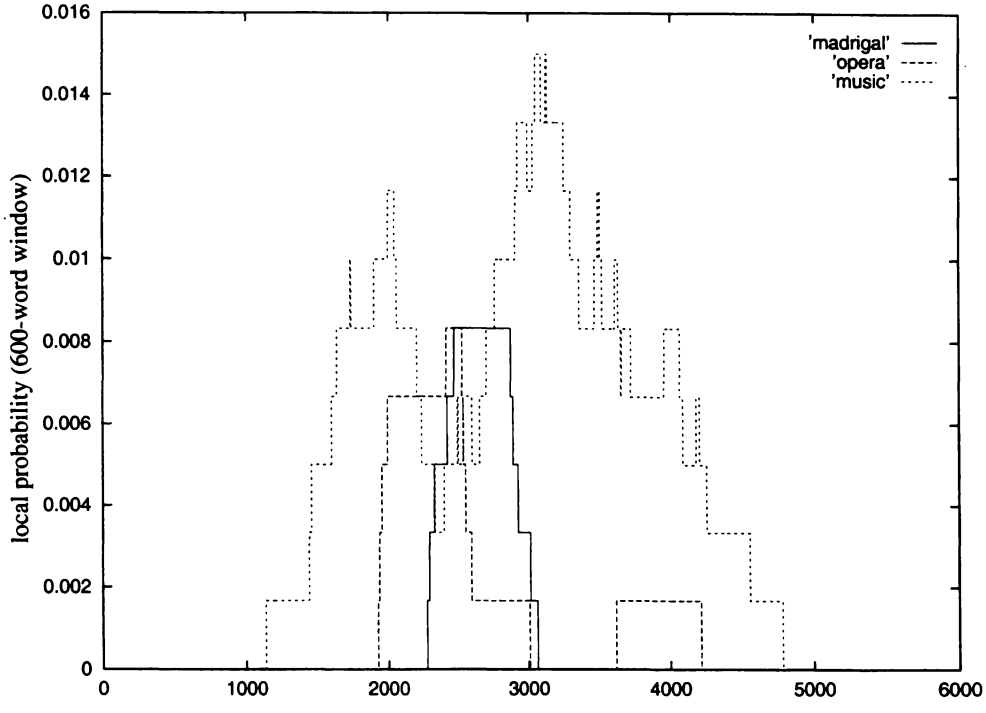


Figure 3.8: Broad semantic latency of "music".

While MTF codes can (under certain circumstances) make gains from semantic latency, the scheme does not extend to take advantage of the more general effects of semantic association between any related pair of content words. A more effective technique would be one that has access to specific statistics for lexical cooccurrences—one that allows words to be predicted from whichever preceding term offers the best semantic cue and therefore delivers the best joint probability.

3.2.3 Mutual information

The notion of a lexical relationship, as defined by both LFG and link grammar, is that a word may project syntactic constraints onto one or more other words in the expression. The implication is that selection over the vocabulary is altered in some way whenever a word introduces such constraints. In the absence of a grammatical theory, it is not clear which subsequent word in the expression is expected to fulfil an outstanding requirement. What

is clear, however, is that the point at which the constraint is satisfied will likely be accompanied by a favourable distortion in the distribution over the vocabulary—favourable, that is, for whichever word satisfies the syntactic constraint.

While LFG and link grammar focus on grammatical relations, semantic latency observes that terms related in meaning often exhibit locality of reference effects in close proximity to each other. Semantic dependency between two words is therefore also likely to produce a conspicuous cooccurrence statistic that can be translated into an improved probability estimate.

These expectations suggest a strategy for detecting unadorned lexical relationships: for each word in the expression, ascertain which preceding term gives the best joint probability. More formally, for each pair of words (w_i, w_j), where i and j are word indices in a sample text, find the index i satisfying $i < j$ that maximises the conditional probability $\Pr[w_j|w_i]$, and thereafter assume a lexical relationship between w_i and w_j. Whether the relationship is syntactic or semantic (or neither) is largely inconsequential. What is important is that any positive difference between "the product of the independent probabilities for w_i and w_j" and "the product of the independent probability for w_i and the conditional probability for $[w_j|w_i]$" implies redundancy in the information content of the related terms—redundancy that can be translated into a reduced complexity estimate for the language sample.

From a communications theory perspective, such redundancy is called *mutual information* [3] and is usually expressed in terms of bits saved [75]. Recall from Chapter 2 that the information content of a symbol i is equal to the number of bits required to optimally encode it, and this is given by the formula $-\log_2 p_i$ bits, where p_i is the probability of symbol i. The independent probability of a word is of course the ratio of the number of times it is observed with respect to the total number of words observed in the sample text.

For example, the word "police" occurs 156 times in the 1,065,795 words of the Brown Corpus, and its information content within a unigram model of that corpus is thus $-\log_2(156/1065795) \approx 12.74$ bits. The word "department" occurs 225 times in the Brown Corpus, and its unigram information content is $-\log_2(225/1065795) \approx 12.21$ bits. The total information content

of the sequence "police department" under the independence assumption is simply the sum of the information content of the two words, about 24.95 bits. However, of the 156 times "police" occurs in the Brown Corpus, the word "department" occurs 5 times immediately after it. For a bigram model, the information content of "department" given "police" as prior context is only $-\log_2(5/156) \approx 4.96$ bits, and the information content of the sequence "police department" is now just $12.74 + 4.96 = 17.70$ bits, giving a savings of $24.95 - 17.70 = 7.25$ bits over the unigram model. This difference is the mutual information of the two words, as given by bigram statistics.

Rosenfeld [90] observes that mutual information is symmetric for a pair of words (w_i, w_j) in that it is the same regardless of whether one uses the forward conditional probability of w_j given w_i or the backward conditional probability of w_i given w_j. For example, "police" precedes the 225 occurrences of "department" 5 times, and its information content is consequently $-\log_2(5/225) \approx 5.49$ bits. The combined information content of "police department" is now $12.21 + 5.49 = 17.70$ bits, giving a savings of $24.95 - 17.70 = 7.25$ bits over a unigram model—the same as before. Thus the mutual information depends only on the pair of words involved and not on the direction of the assumed dependency. This is useful when it comes to establishing lexical relationships in that mutual information can be taken as a measure of undirected *lexical attraction* between two words. Under the assumption that pairs of words with a genuine linguistic relationship will demonstrate a high level of lexical attraction, mutual information provides a basis for hypothesising about which word in an expression satisfies a lexical constraint introduced by another.

3.2.4 Trigger pairs

Because mutual information for two words depends directly on their joint probability, precisely how that probability is established greatly affects the level of lexical attraction that can be assumed. The example given above relies on the statistics of bigram models, thus the conditional probabilities pertain to adjacent words. In fact, the bit savings obtained from the bigram models in Chapter 2 are simply measures of the mutual information for pairs of adjacent terms. But mutual information as given by bigram statistics is

of little value for establishing lexical relationships because adjacent words are assumed to be linked in n-gram models anyway, regardless of how strong their lexical attraction. For mutual information to be more useful, a more flexible model of language structure is required—one able to consider more distant dependencies. That is, the model must be able to search back in the sequence for the word most strongly *attracted* to the current one. This requires access to more general cooccurrence statistics for word pairs.

Rosenfeld [91] describes a *trigger pair* as a pair of words, $(s,t)_H$, for which the probability of t is given a "boost" if the word s appears somewhere in the history H, where the history is a fixed-length window of words immediately preceding t. The boost is of course equal to the mutual information for s and t, and this leads to a slightly more rigorous definition for mutual information:

$$\mathrm{MI}(s,t)_H = -(\log_2 \Pr[t] - \log_2 \Pr[t|s]_H)$$

where $\Pr[t|s]_H$ is the probability of t given that s occurs earlier in history H, and thus $\mathrm{MI}(s,t)_H$ is the mutual information for s and t in that history. In this formulation, an increased local probability from semantic latency is simply the boost afforded by a self-trigger pair (s,s). To maximise mutual information, however, a more exhaustive search of trigger pairs is necessary.

3.2.5 The history length trade-off

The formula for mutual information shows that the degree of boost is in part a function of how far back the model is prepared to look when calculating lexical attraction (i.e. the length of H). Obviously if there is to be any lexical attraction for a pair of words they must both appear in the history. But as the length of the history grows without bound, $\Pr[t|s]_{|H|}$ given any s must approach $\Pr[t]$, and the mutual information for s and t accordingly approaches zero. More generally, prior occurrence of the conditioning word has an ever diminishing effect on the likelihood of another word as the distance between them increases. Beeferman *et al.* [7] provide empirical evidence that the boost for t decreases exponentially as the length of H increases, becoming more or less constant (and insignificant) at around $|H| = 400$.

A sequence model that aims to capitalise on mutual information must manage the trade-off between the availability of mutual information and its magnitude with respect to the length of the history. That is, if $\Pr[w_i, w_j]_H$ is the probability that w_i and w_j cooccur in a history of length H, and $\mathrm{MI}(w_i, w_j)_H$ is the corresponding mutual information, then an optimal mutual information model for a sample of length N, based upon a fixed length history, would compute

$$H = \mathrm{argmax}_{2 \leq h \leq N} (\sum_{i=1}^{h-1} \sum_{j=i+1}^{h} \Pr[w_i, w_j]_h * \mathrm{MI}(w_i, w_j)_h).$$

For most language samples, such a computation would be, if not intractable, at least impractical. Huang *et al.* [55] undertook an investigation into the effects of lexical distance on mutual information through an extensive study of *long-distance bigrams.* Recall that a conventional n-gram model uses $n-1$ immediately preceding words as the context for predicting the next word. A long-distance n-gram uses $n-1$ words some distance back as the predictive context. For example, a distance-2 trigram predicts w_i on the basis of (w_{i-3}, w_{i-2}). Thus a distance-1 bigram is a conventional bigram. Huang *et al.* systematically explored the amount of information in long-distance bigrams from distance-1 to distance-1000 (under the assumption that there would be no significant information at or beyond this extreme distance). Their results indicate that there is significant information in the last five words of history, but that average mutual information at greater distances is virtually neglible. More importantly, they conclude that long-distance n-grams are seriously deficient for language modeling because they fail to merge instances of correlation at different distances—that is, in order to take advantage of mutual information at a distance, a model must be able to generalise over variable length histories.

Given that mutual information for word pairs appears to peak within a history of about five words, and that generalisation over variable length histories seems to be essential, there is a natural assumption about syntactic dependencies that implies a very simple heuristic for exploiting short but variable history lengths: simply restrict the search to word pairs within the bounds of a single sentence.

Webster's Revised Unabridged Dictionary defines a sentence (grammatically) as "a combination of words which is complete as expressing a thought." Completeness of thought suggests an implicit cohesion between its words—or, more plainly, given that it *is* a sentence, its words are de facto related. Combined with observations from long-distance n-grams, the implication is that sufficiently high levels of lexical attraction are likely available within the bounds of a single sentence, and that extending the search for lexical dependencies beyond this is unwarranted. Indeed, maximising mutual information over sentence-bounded word pairs forms the basis for yet another in the exponential family of stochastic lexical models—the so-called lexical attraction model.

3.3 Lexical attraction

The lexical attraction model is an entropy-based scheme for assigning structure to isolated sentences. It links together words with high mutual information on the assumption that genuine syntactic relations result in high joint probabilities. A complete syntactic structure is derived by finding a minimum set of lexical links defining a connected acyclic planar graph for the entire sentence (where each word is a vertex), such that the total mutual information for the sentence is maximised.

Like n-gram models, the lexical attraction model specifically focuses on the information theory perspective of language in that its goal is to capitalise on strong joint probabilities for related pairs of words. Unlike n-gram models, however, the cooccurrence statistics are not restricted to adjacent terms.

3.3.1 Entropy and syntactic relations

Following standard structural models, the lexical attraction model insists that every word in a sentence be attached to a dependency structure in accordance with syntactic relations. Unlike traditional structural accounts, however, the lexical attraction model has no predefined syntactic relations available to make judgments about where attachments should go. Instead, it infers them on the basis of mutual information—that is, lexical attraction

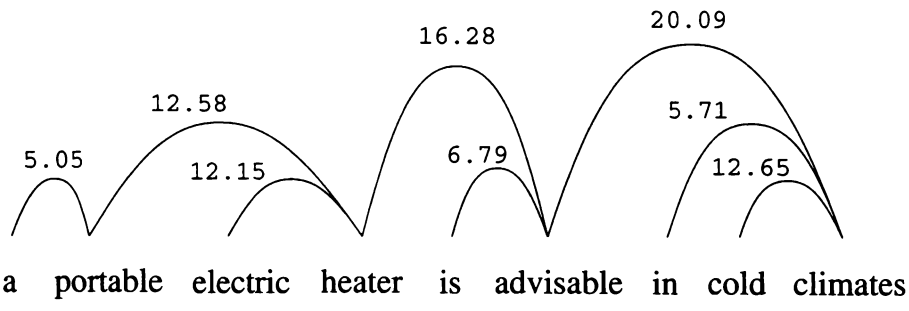


Figure 3.9: A lexical attraction dependency structure.

is taken to be the likelihood of a syntactic relation and is therefore used to decide where links should go.

Recall that the information content of a word can be estimated as its negative log likelihood under the independence assumption, and that the information content of a sentence is therefore simply the sum of the information content of its words. For example, consider again the Brown Corpus sentence from Section 2.2, annotated with the information content of each word.

a	portable	electric	heater	is	advisable	in	cold	climates
5.51	16.32	13.89	16.22	6.72	20.02	5.64	12.58	20.02

Based on unigram statistics, the total information content of this sentence can be expressed in 116.92 bits. But if the probability of each word is made conditional on the context of the immediately preceding term, the (usually) higher joint probabilities lead to a decrease in the information content of each word, as with

a	portable	electric	heater	is	advisable	in	cold	climates
5.60	11.71	2.70	4.54	3.81	13.30	3.64	12.38	7.44

where the total bigram information content of the sentence is now only 65.12 bits—a savings of 51.8 bits. (Recall from Section 2.1.5 that end-of-sentence markers are treated as individual words in this study, thus the conditioning context for the first word in the sentence above is a preceding fullstop.)

As noted earlier, the savings obtained when one word is predicted based on the context offered by another is the mutual information for the pair—a symmetric relation. Because it is not always true that the immediately preceding word offers the best predicting context, higher mutual information may be found through examination of cooccurrence statistics for all pairs of words in the sentence. The goal of a lexical attraction model is to find a set of lexical links that ties all the words together as a single dependency structure in such a way as to maximise the available mutual information. Figure 3.9, for example, shows a lexical attraction dependency structure for the sample sentence, annotated with the mutual information of each syntactic relation. This structure estimates the total mutual information of the sentence at 91.3 bits—almost double the 51.8 bits saved using bigrams.

3.3.2 Dependency structure

In the lexical attraction model, every word in a sentence must be linked to at least one other to produce a single dependency structure. Because any word can be used as the context for predicting another, there is potential for much higher joint probabilities than what is possible from n-grams. However, unlike n-grams, it is not obvious which word is predicting another in a lexical attraction account, and the uncertainty of the dependency structure must therefore be factored into the entropy calculation for the sentence. Formally, the entropy of a sentence S with dependency structure L is calculated as

$$-\log_2 \Pr(S) = -(\sum_{w_i} log_2 \Pr(w_i) - \sum_{(w_i,w_j)\in L} \log_2 \frac{\Pr(w_i, w_j)}{\Pr(w_i)\Pr(w_j)} + \log_2 \Pr(L))$$

In other words, the information content of the sentence is the sum of the information in the words, less their mutual information from syntactic relations, plus the information in the dependency structure.

The n-gram model itself assigns a dependency structure to a sentence; one where each word is linked to the preceding word. But the n-gram structure is implicit and can be predicted with perfect accuracy; thus it adds nothing to the entropy of the sentence. The potential mutual information savings from

lexical attraction, however, can only be realised if the cost of defining the dependency structure is less than the difference between the mutual information available from lexical attraction and the mutual information captured through n-grams.

Yuret [114] notes that if selection of links is unconstrained—as when each word is allowed to be linked to whichever other word gives the highest measure of mutual information, without regard to planarity—then the number of possible dependency structures for a sentence with n words is n^{n-2} (as per Cayley's formula [53]) and encoding of the structure requires about $n \log n$ bits. With the planarity constraint, the situation is much better. Yuret observes that encoding of planar dependency structures is linear with respect to the length of the sentence—approximately 2.75 bits per word.

3.3.3 Structure assignment

Given the potential for efficient encoding of the dependency structure, it is still necessary to devise some method for assigning structures in the first place. Yuret [114] demonstrates how a Viterbi style algorithm can be constructed to find the dependency structure with the highest mutual information. A dependency structure is a non-rooted tree and, under the planar constraint, each node of the tree corresponds to a contiguous subsequence of words (i.e a *span*). Each span is linked into the complete dependency structure by a syntactic relation between one of the words within the span and one outside it. By decomposing a sentence into shorter and shorter spans, it is possible to devise a recursive computation that builds an optimum dependency structure bottom up, starting with shortest spans and combining them into longer ones.

Yuret shows that computation of an optimum solution in this way can be done in $O(n^5)$ time. But, because his goal is to develop lexical attraction as a mechanism for learning syntactic structure, he argues that the runtime of the optimal algorithm is excessive. He proposes a much simpler approximation procedure that runs in $O(n^2)$. As each new word is encountered, it is linked to whichever preceding word shows the highest mutual information. If the link produces a cycle or a crossed link, the weakest of the links in conflict is eliminated. The process continues left to right through the sentence, linking

and unlinking words in such a way as to maintain at all times a planar graph for all words that have been seen. Though the resulting structure may not be optimal, Yuret's experiments indicate that the marginal gain of the optimal algorithm is not significant.

3.3.4 Lexical attraction in practice

Bach and Witten [5] experimented with lexical attraction as a model for text compression. Using Yuret's approximation algorithm, they attempted to devise a system of representation for efficiently encoding the dependency structure. Yuret had proposed an enumeration of all possible planar structures for a sentence of n words, allowing the actual structure to be encoded using its index in that set—as noted earlier, a technique that requires 2.75 bits per word. Bach and Witten improved on this by instead encoding the number of forward and backward links for each word. The pairs of link-counts are treated as single symbols and compressed in isolation using a first order Markov model. The results indicated an average overhead of between 1.61 and 2.38 bits per symbol over four large texts.

Empirical studies by Bach and Witten on the 1.2 million words of *Jefferson the Virginian* show that a lexical attraction model can deliver gains in lexical entropy of 24% over a unigram model, in comparison to 16% savings available from bigrams—a considerable improvement given that their model is entirely adaptive, estimating distributions on the fly as more and more of the text is encoded. However, they concede that this result is misleading in that it does not include the overhead of encoding the graph structure. Despite their improved method of structure representation, the total entropy of the data when structural information is included is considerably worse than estimates obtained from conventional bigrams.

Bach and Witten traced the deficit directly to the overhead of encoding the dependency structures. But this should not be interpreted as conclusive evidence against the viability of lexical attraction models in general. As an adaptive lossless compression scheme, their model had to confront practical issues such as actually encoding words and link-counts explicitly the first time they are encountered—details normally overlooked in theoretical discussions of lexical attraction. Such overheads diminish as more and more text is

compressed, but only fade to nothing as the text size approaches infinity. The effects can be mitigated to some extent through prior training on a baseline text. However, certain characteristics observable in dependency structures suggest an entirely different solution: a simpler approximation algorithm that permits access to much of the mutual information available from lexical attraction without incurring any cost in the dependency structure. The idea is to alter the default structure assigned by n-gram models so as to increase the incidence of assumed adjacency between words likely to show strong lexical attraction—specifically, the content words.

3.3.5 The dominance of semantic relations

Even though there is no implicit requirement to capture bona fide linguistic structures in a lexical attraction model, there is an underlying assumption that actual lexical relations exhibit stronger lexical attraction than unrelated pairs. Real lexical relations are primarily either syntactic or semantic. For example, in the sentence "the boy is eating a banana" we may say that there is an obvious semantic relationship between "eating" and "banana" in the sense that bananas frequently get eaten. We may further say that there is a similar semantic relationship between "boy" and "eating", given that boys eat while, say, rocks do not. Any assumption of a semantic relationship implies availability of mutual information from conditional probabilities—in this case, of the form Pr[eating|boy] or Pr[banana|eating].

In comparison, the relationship between "the" and "boy" is qualitatively different in that their semantic association is, if not absent altogether, at least very much more abstract. That is, seeing the word "the" does not so much present a semantic cue that specifically increases the probability of "boy", rather it provides the syntactic cue that a noun is imminent. That the noun is "boy" instead of "rock" is not in any way implied by the determiner. Similarly, the auxiliary "is" provides a syntactic context for increasing the probability of an impending verb, but offers no semantic hint that the verb will likely pertain to the usual behaviours of a boy.

Lexical attraction models do not differentiate between different types of lexical relations. Nevertheless, the difference between semantic relations and syntactic relations does manifest itself in dependency structures. Due to the

relative infrequency of content words, semantic relations tend to yield much higher levels of mutual information than syntactic relations. As a consequence, the tendency of structural inference algorithms is to prefer semantic relations when making decisions about which links to preserve. Syntactic links involving function words, on the other hand, typically offer only a fraction of the savings in comparison with semantic relations, and are often tacked on wherever they can be so as to complete the graph without violating the planarity condition.

The result of this preferential status for semantic relations is a tendency for content words to be linked to each other regardless of how far apart they are, and for function words to be linked to adjacent terms regardless of lexical category. Not surprisingly, dependency structures often bear a striking similarity to conventional grammatical structures. In the latter, top level syntactic structures link constituent substructures, each of which is normally headed by a word from a semantic lexical class. Substructures that have more than one content word will often link these first, while functional elements are usually added as ancillary features at the lowest levels.

3.3.6 An alternative approximation algorithm

The tendency for semantic relations to dominate lexical dependency structures suggests an alternative approximation procedure for choosing links—one that can preserve the bulk of mutual information savings with no cost entailed by the structure itself. If content terms are regarded as a separate stream of words, such that two content words are taken to be adjacent (or *super-adjacent*) when there are no other content words between them, then a bigram model constructed for this stream alone would link them implicitly. Given that close proximity content words typically demonstrate strong lexical attraction, their mutual information is utilised. But, by garnering bigram statistics for super-adjacent content words, the incidence of many conventional semantic bigrams increases. For example, "boy was eating", "boy is eating" and "boy has been eating" all contain the same super-adjacent content-word bigram ("boy", "eating"). The net effect is fewer bigrams, more reliable statistics, and greater utilisation of high mutual information. Moreover, given that the planarity constraint for dependency structures prevents

links between pairs of content words from crossing, super-adjacent content words experience greater preference in conventional lexical attraction models anyway, and this is preserved.

Given that relationships involving function words are largely syntactic (pertaining more to associations between lexical classes than explicit words), function words could be predicted from an adjacent term regardless of its class. Moreover, since the relationship between a function word and a content word is predominantly class-based, recording the content word explicitly is somewhat unnecessary. That is, content words could be reduced to a class symbol without a significant penalty in terms of loss of mutual information in their relationship with an adjacent function word. By treating that class symbol as a functional term, all bigrams involving function words can be reduced to pairs of functional elements. Given that the set of function words is quite small, a comprehensive set of function word bigrams can be constucted more quickly and utilised more frequently than in a conventional bigram model.

If all content words are predicted on the basis of the most recent prior content word, and if all function words are predicted on the basis of the most recent prior function word (or content class symbol), then the entire dependency structure is implicit and its entropy is zero. At the same time, a super-adjacency bigram model of this kind would be smaller and more compact, allowing more reliable statistics to be obtained more quickly, thereby reducing the amount of data needed for conditioning accurate probability estimates. Furthermore, a substantial number of links from an optimum dependency structure are likely to be preserved by a super-adjacency model, allowing mutual information to contribute to reduced complexity estimates for language.

3.4 Discussion

Finite context models fail when the context does not contain useful information. Since allowing the context to grow without bound leads to intractably large models, an alternative approach is to allow the context to move. That is, instead of restricting context to immediately preceding words, allow any

preceding word to be used for estimating the probability of the next word. More to the point, use the preceding word which maximises that probability.

Allowing probabilities to be assigned to relationships between nonadjacent words is precisely what a stochastic link grammar attempts to achieve. Given that the relationship between, say, a determiner and noun is genuine, link grammars characterise the association with a syntactic link, and stochastic versions assign a probability to the relationship. But the link formalism requires that words be annotated with the grammatical information to ensure that a given linkage is valid—an unnecessary overhead for entropy-based sequence modeling. Lexical attraction models, on the other hand, establish lexical relationships solely on the basis of mutual information. Unfortunately, the entropy entailed in specifying which words are linked tends to wipe out any gains afforded by improved joint probabilities.

Whether a relationship between two words within a sentence is predominantly semantic or syntactic is inconsequential to the notion of lexical attraction, but the fact that there are two distinct forms of lexical dependency suggests two layers of language structure interlaced within a stream of words. One layer comprises highly meaningful terms played out in a sequence dictated by a temporal stream of discourse semantics. Adjacent terms in this stream are likely to contain very high levels of mutual information. The other layer comprises the grammatical terms needed to satisfy the syntactic requirements of language. Mutual information for pairs of words in this stream arise from syntactic dependencies and are thus not as susceptible to the effects of semantic latency as meaningful terms.

It is hypothesised that a parallel bigram model that maintains separate statistics for these two streams would preserve most of the lexical links of an optimum dependency structure without incurring any cost for specifying those links. This is the premise of the super adjacency models described in Chapters 5 and 6. However, given that interaction of these two streams is coordinated by an assumption about the relationship between lexical classes in language, it is useful first to examine the effects of class-based sequence modeling more closely in the next chapter.

Chapter 4

Category-based Models

Traditional theories of syntax usually express grammatical relations in terms of lexical categories, not as direct links between specific words. In Phrase Structure Grammar, for example, a typical nounphrase might be described as "a determiner followed by zero or more adjectives followed by a noun." Given that determiners do (under normal circumstances) signal imminent arrival of a noun, class dependencies of this sort appear to be genuine and can be exploited to improve the predictive capacity of a statistical model.

This chapter explores methods for translating class information into improved complexity estimates by entropy-based models. Underlying assumptions of category-based lexical prediction are outlined, along with summaries of existing techniques that use fixed- or variable-length part-of-speech contexts. It is argued that such methods have the potential to give much better complexity estimates than conventional word-based approaches because the relative compactness of their models mitigates the problem of data sparseness, reducing the amount of data required to condition accurate probabilities.

Class-based entropy models depend heavily on prior knowledge of lexical categories before structural inference can commence—knowledge most often provided by automatic tagging systems which are themselves class-based entropy models trained on previously tagged texts. To break into this circle, an initial classification scheme is needed. A series of experiments is conducted with results that suggest part-of-speech information need not be terribly accurate to yield benefits for stochastic modeling. It is argued that a very

crude classification scheme based upon the distinction between open- and closed-class words is largely sufficient for exploiting categorial dependencies with class-based n-grams. This is a key component of the super-adjacency models outlined in the next chapter.

4.1 Category-based lexical prediction

A number of stochastic modeling schemes have attempted to capitalise on the additional linguistic regularity accessible when lexical categories are available to the predicting mechanism. The underlying idea is that knowledge of a word's category improves a model's ability to predict it, and that the category itself can be predicted solely from its class-based context. This section outlines some of the fundamentals of category-based stochastic modeling. It is shown that while the underlying principle can lead to improved lexical predictions, this cannot be translated into gains in overall complexity estimates without a significant amount of prior linguistic knowledge.

4.1.1 Weak structural assumption model

In a word-based unigram model, the length of the code for a word is proportionate to the negative log of its independent probability—specifically, as per Shannon's formula, the length of the code in bits for a word w with probability $\Pr[w]$ is equal to $-log_2\Pr[w]$. Knowing the category of the word changes the basis for its prediction *from* merely its global frequency with respect to the complete vocabulary *to* its frequency with respect to the subset of the vocabulary comprised of just those terms from the same lexical class. More plainly, the probability for the word becomes conditional on knowledge of its lexical category; thus the code length for a word w, given its part of speech tag t, is equal to $-log_2\Pr[w|t]$.

The *weak structural assumption* made by category-based prediction is that the probability of occurrence for a word is assumed to be dependent solely upon the category to which it is believed to belong [84]. Of course, the penalty for models based on this assumption is that the category information itself must somehow be included in the final complexity estimate for the word. Like the word-based unigram model, the code length for a tag is proportionate to

its negative log likelihood under the independence assumption—that is, the length of the code for a tag t is equal to $-log_2\Pr[t]$. The code length for a word encoded under the weak structural assumption is thus the sum of the code length for its tag plus the code length for the word given its tag, or more formally

$$-(\log_2 \Pr[t] + \log_2 \Pr[w|t])$$

Leaving aside the problem of how the tag can be known, the immediate question is whether knowledge of the tag in such a simple model actually buys anything for encoding a word.

4.1.2 Binary categories

In a word-only unigram model, each bit of a word's code can be thought of as a kind of binary category symbol, partitioning the vocabulary into two subsets according to whether or not the bit in question is set or clear. For example, the most significant bit differentiates between all words whose code begins with a one as opposed to a zero and selects the subset with the matching high-order bit. Each subsequent bit continues in this way to subdivide the set of possible words into those that may contain the word in question and those that may not, until the last bit uniquely identifies that word. The total number of bits required is proportionate to the negative log likelihood of the word. While this view assumes integral bit codings for each word, the principle applies to other coding schemes as well, albeit in a more complex way.

In a tag-based model based on the weak assumption, the bits of a tag's code subdivide the set of tags in the same way. Once the tag is known, some subset of the vocabulary which includes the target word has implicitly been isolated in a similar manner to what is achieved using the first few high-order bits in the word-only unigram code. Though the number of additional bits needed to identify the target word is reduced by the increased probability afforded by the conditioning context of its tag, the savings are exactly equal to the number of bits required to identify the tag beforehand, resulting in no net gain.

tag	$\Pr[t_i]$	word	$\Pr[w_j]$	$\Pr[w_j \mid t_i]$	$2^{(\log_2 \Pr[t_i] + \log_2 \Pr[w_j \mid t_i])}$
t_1	1/4	w_1	1/5	4/5	1/5
		w_2	1/20	1/5	1/20
t_2	3/4	w_3	1/2	2/3	1/2
		w_4	1/4	1/3	1/4

Table 4.1: Probabilities for tags and words of a small language.

The situation is perhaps easier to see with a simple example. Consider Table 4.1 showing probabilities for the tags and words of a small language; one that has just two categories with two words in each. In this language w_1 and w_2 are words of type t_1, while w_3 and w_4 are of type t_2. The independent probabilities for each type are given in the second column, and the independent probabilities for the words are given in the fourth column. Column five lists the conditional probabilities for each word given its type. A tag-based context model must encode each word as a pair—the code for its tag and the code for the word given its tag—thus the length of the code for the pair is equal to the sum of the lengths of the codes for each component, which is equal to the negative log of the product of their probabilities, and the inverse of this is given in column six of the table. A comparison of column six and column four shows that the code length for a tag/word pair is the same as for the word by itself when its category is not known.

4.1.3 Class ambiguity

The result from the example given above is not surprising because the part-of-speech label carries no information not implicit in the word itself. That is, given the word, the category is immediately known. Because the number of tag-word pairs is exactly the same as the number of words, the probability of any one tag/word pair is obviously the same as for the word alone, thus the overall complexity estimate for that word is the same with or without knowledge of its lexical category.

In natural language, however, words typically belong to more than one category and the global probability for a word is distributed over its set of possible tags. If tag/word pairs are encoded independently, their complexity

tag	$\Pr[t_i]$	word	$\Pr[w_j]$	$\Pr[w_j\|t_i]$	$-\log_2 \Pr[t_i] - \log_2 \Pr[w_j\|t_i]$
t_1	3/8	w_1	1/5	8/15	1/5
		w_2	1/20	2/15	1/20
		w_3	1/8	1/3	1/8
t_2	5/8	w_3	3/8	3/5	3/8
		w_4	1/4	2/5	1/4

Table 4.2: Probabilities for a small language with class ambiguity.

estimates must degrade as type ambiguity increases because the number of distinct pairs gets bigger—effectively increasing the number of parameters over which the distribution must be approximated.

Consider the situation when the type t_2 word w_3 from the earlier example also belongs to category t_1 one quarter of the time. The distribution for this new language is shown in Table 4.2. The global probability of w_3 is unchanged from the previous example, and its conditional probability still improves over its independent probability when it is tagged as t_2. Its conditional probability when tagged as t_1, however, entails an increased cost in its code length in comparison to coding it with respect to its independent probability alone. In fact, its average expected complexity almost doubles. When w_3 was unambiguously of type t_2 its complexity was constant at $-\log_2 1/2 = 1$. Under the new categorisation scheme, a quarter of the time its probability is 1/8 and three quarters of the time it is 3/8, giving an average complexity of $1/4 \times (-\log_2 1/8) + 3/4 \times (-\log_2 3/8) = 1.81128$—a signficiant increase directly attributable to the ambiguity of its category.

As noted in Chapter 3, most language samples exhibit highly nonstationary statistical characteristics, such as a varying local probability for a given word within a finite window. The same variation may be observed for a word being used in one of its possible class senses. For example, the word "hit" in an American baseball commentary may be used more often as a noun than as a verb, while the reverse is perhaps more usual in common parlance. An adaptive model able to condition its tag-based predictions in accordance with local statistics can use local preferences to achieve improved complexity estimates for finite language samples.

Consider, for example, the altered distribution for w_3 shown in Table 4.3.

tag	$\Pr[t_i]$	word	$\Pr[w_j]$	$\Pr[w_j\|t_i]$	$-\log_2 \Pr[t_i] - \log_2 \Pr[w_j\|t_i]$
t_1	7/20	w_1	1/5	4/7	1/5
		w_2	1/20	1/7	1/20
		w_3	1/10	2/7	1/10
t_2	13/20	w_3	2/5	8/13	2/5
		w_4	1/4	5/13	1/4

Table 4.3: A more favourable distribution for the ambiguous class.

In this scenario, w_3 is used in sense t_1 one fifth of the time and in sense t_2 four fifths of the time, giving an average complexity of $1/5 \times (-\log_2 1/10) + 4/5 \times (-\log_2 2/5) = 1.72193$—a net improvement from the previous example. Thus a net savings can be realised by a tag-based model when within a finite text an ambiguous term is used in a particular sense with greater probability than is expected in a more representative sample of the language. Under any other circumstances, however, the ambiguity of the category increases the entropy of the sample, leading to poorer complexity estimates than would be obtained from a strictly word-based model.

4.1.4 Class-based n-grams

The principal limitation of the weak structural assumption is its failure to make proper use of the way in which lexical classes demonstrate regularity in language. Conventional theories of syntax describe linguistic structure in terms of part-of-speech categories, not in terms of explicit patterns of words. For example, a nounphrase is typically defined as a sequence comprised of a determiner, some adjectives and a noun, with similar definitions for the verbphrase, prepositional phrase and so forth. The fact that determiners always mark the onset of a nounphrase means that occurrence of a determiner can be used to trigger a local probability boost for the noun class. Similarly, appearance of a modal auxiliary typically signals the start of a verbphrase and can thus be used to temporarily assign a higher probability to all verbs until one is seen. More generally, the context afforded by occurrence of a particular word type can momentarily distort the probability distribution over the set of types in such a way as to temporarily favour the category

of the next word. Even if the entropy of a tag/word pair is equal to the entropy of the word by itself, as is argued in the preceding section, more accurate prediction of the tag leads to a net reduction in the entropy of the pair. This suggests the use of class-based contexts as a means for improving the predictive capacity of an entropy-based language model.

Brown *et al.* [18] explore the possibility of achieving good complexity estimates for language using class-based n-grams. They combine the weak structural assumption with an additional assumption that the probability of a category is dependent solely on its category n-gram context. Formally, they calculate the probability of a word as

$$\Pr[w_i] = \Pr[w_i|t_i] \times \Pr[t_i|t_{i-k}, \ldots, t_{i-1}]$$

where the probability of the i-th word, w_i, is computed as the product of the conditional probability of w_i given its tag, t_i, and the conditional probability of that tag given the context of the previous k tags.

The category n-gram context used to predict the current tag is defined by the sequence of tags associated with the $n-1$ words immediately preceding the current word. More accurately, then, the probability of the next tag is approximated for an equivalence class s given that the word history $(w_{i-k}, \ldots, w_{i-1})$ belongs to s, where $k = n-1$. As any particular word history may belong to several equivalence classes, the model must take into account the probability that the equivalence class for the history in question is indeed s. Thus, the formula for predicting the tag may be further decomposed into

$$\Pr[t_i|(w_{i-k}, \ldots, w_{i-1})] = \Pr[t_i|s] \times \Pr[s|(w_{i-k}, \ldots, w_{i-1})].$$

Because there are far fewer unique equivalence classes than there are different word histories of the same length, the model is able to condition an accurate estimate for $\Pr[t_i|s]$ quite quickly. That is, given a vocabulary of size V and a tag set of size T, an order-3 category-based n-gram model has at most $T \times (T^2 + V)$ independent parameters, whereas a word-only model has closer to V^3 parameters. As T is often quite small (typically around one or two hundred tags) and V is usually quite large (often many tens of thousands of words) the category-based approach entails considerably fewer contexts than the word-based approach, resulting in a much more compact

model. This means that the counts for contexts observed in the language sample will be substantially less sparse for a category-based model, allowing accurate statistics to be formulated more quickly than a comparable word-based model.

4.1.5 Fixed-length class contexts

Brown *et al.* [18] experimented with class-based trigram models using a vocabulary of 260,741 words garnered from a 365,893,263 word training text composed from a variety of sources. The vocabulary was partitioned into 1000 distinct nonoverlapping categories based on a mutual information heuristic. After training, they observed that the class-based model required just one-third as much storage as a word-based trigram model.[1] They report, however, that the complexity estimates from their class-based model were slightly worse than from the conventional word-based model—though, to be fair, their experiments were directed primarily at the prospects of using such models as a basis for class inference rather than as a specific means for reducing overall estimates of entropy.

Teahan and Cleary [111] applied the concept of category-based n-grams to text compression with the explicit goal of trying to minimise estimates of language complexity. Their scheme uses two parallel models, each of which employs a blended context comprised of the previous word and some number of preceding tags. Specifically, the estimated probability of the next word is based on a trigram model, where the conditioning context consists of the word's tag and the preceding word. The tag itself is predicted using a 4-gram model, where the context consists of the preceding word and its tag and the tag before that. Probabilities are formed adaptively and, to be robust, each model escapes to a lower-order (ultimately character-based) context when the next word has not been seen in the current context. Despite this more sophisticated use of class contexts, their empirical studies with conventionally tagged versions of the LOB Corpus and Wall Street Journal produced results almost identical to those from standard word-based bigrams: at best 0.001 bits per character better on the LOB Corpus and at worst by 0.008 bits per

[1] If both models were exhaustive, then the class-based model would be nearly eight orders of magnitude smaller.

character poorer on the Wall Street Journal. It is worth noting, however, that this comparable performance was achieved from a considerably smaller model.

4.1.6 Variable-length class contexts

Because there are far fewer possible class-based contexts than word-based ones, a class-based n-gram model is more compact than a conventional model, and n-gram counts in the training data are less sparse. Nielser and Woodland [84] observe that these properties make the use of deeper contexts (say, longer than three or four preceding symbols) feasible both in terms of memory requirements and efficient calculation of probability estimates. They propose that the availability of longer contexts could decrease the amount of class ambiguity arising from polysemy, allowing more accurate prediction of lexical tags.

They developed a statistical part-of-speech tagging algorithm which makes use of a multiway trie for efficiently storing variable length equivalence classes comprised of part-of-speech traces gleaned from a tagged training text. To avoid the potential for excessive growth of the model arising from unbounded contexts, they employ a pruning criterion which retains an equivalence class only if it decreases the average log likelihood of the training corpus by a predefined threshold, a value derived from *ad hoc* experiments using Duda and Hart's leave-one-out cross-validation framework [39].

Surprisingly, Niesler and Woodland's so-called *varigram* tagger does not perform significantly better than Elworthy's bigram based ACQUILEX tagger [42] for overall tagging accuracy. However, their algorithm does do considerably better at tagging *out of vocabulary* words. Because their n-grams express regularities in terms of lexical categories, the varigram model is able to generalise over unseen word sequences. When the category of a new word is known, or can be guessed, it inherits the n-gram statistics observed for that category. Given that the sequential behaviour of the category is likely to be consistent regardless of the exact word in the history, probabilities for novel words can be estimated more accurately from class-based n-grams than from explicit word contexts. Finite class-based contexts have this same property to some extent, but a variable-length context has a distinct advantage. Given

that lexical categories are the substance of grammatical structure, and that such structures may occur with varying lengths, a variable-length class-based model is more likely to have observed an exemplar of the equivalence class required to predict the class of a novel word.

4.2 Class characteristics

A significant drawback for many of the class-based models outlined in the previous section is their *a priori* requirement that the input sample be accurately tagged with part-of-speech labels. This creates the somewhat undesirable situation where a model of language must already be available before another can be inferred. Brown et al. [18] attempted to circumvent this problem by devising a clustering algorithm to create classes heuristically on the basis of mutual information before garnering class-based n-grams, but they concede they were unable to achieve a partition that gave final complexity estimates as good as conventional word-based n-grams.

This section describes a series of experiments devised to examine the importance of accurate classification when seeking to exploit class dependencies in an n-gram model. By deliberately altering the tagging scheme for the input and observing the effect on the complexity estimate from an unrestricted class-based n-gram model, it is possible to form hypotheses about the characteristics of class information that are most useful for predicting language structure. The results indicate that a very crude categorisation scheme based on the distinction between open- and closed-class words is largely sufficient for taking advantage of category dependencies in an entropy-based model.

4.2.1 Unrestricted class-based context

From a linguistic perspective, lexical categories are the material of syntactic structures such as nounphrases and verbphrases. Though appearance of, say, a determiner certainly foreshadows appearance of a noun, in no way can it be used to help predict which of all possible nouns is about to be observed (subcategorisations notwithstanding). The possible benefit of class-based prediction lies with its potential to lower the entropy of tags from a characterisation of syntactic regularity.

N-gram models of syntax, being stochastic regular grammars, are not able to capture the complete grammatical structure of natural language [25]. Jelinek [58], however, points out that the high frequency of a relatively small number of local syntactic constructs in typical language samples gives n-gram models the ability to extract many useful elements of grammatical structure. To that extent, variable-length class-based n-gram models can be satisfactory practical approximations to natural language grammars.

Niesler and Woodland [84] limited the set of retained n-grams in their variable-length class-based model to avoid the problem where complexity estimates continue to increase monotonically as contexts lengthen. By removing this constraint, and allowing class n-grams to grow without restriction, the model can take full advantage of the predictive capacity available in class contexts, mitigating some of the shortcomings that arise from the overall inadequacy of the formalism. This provides a mechanism for studying the effects of various classification schemes on grammatical complexity.

4.2.2 TPD

A greedy tag-phrase dictionary system (TPD[2]) for lossless text compression can be constructed in a manner similar to LZW, a variant of LZ78 proposed by Welch [112]. A tagged-text is given to the encoder as a stream of *word/tag* pairs. The encoder parses the input into a dictionary of phrases, where each phrase is constructed as a trace of lexical tags from the input. The phrase dictionary is maintained as a multi-way trie, initialised to include each unique tag as an individual phrase. As each word/tag pair of the input is read, the encoder finds the longest matching trace for the tag sequence and then outputs the appropriate phrase number.

As each phrase merely depicts a sequence of lexical categories, additional information is encoded with the phrase number to identify the correct word for each tag. When a new word/tag pair is encountered, the word is explicitly encoded for output and then added to the appropriate dictionary as given by the tag. If the word is already in the dictionary, it is arithmetically encoded according to the empirical distribution for that category derived from counts

[2]This system is the result of collaborative work with Ross Peeters, and a report on its compression performance appears in *Proceedings of DCC'98* [102].

maintained for each word. The output from the encoder is thus a variable-length tuple of the form $(\sigma_j, w_1, w_2, ..., w_n)$, where σ_j is the encoded phrase number and w_i is the encoded word for the i-th category symbol in σ_j.

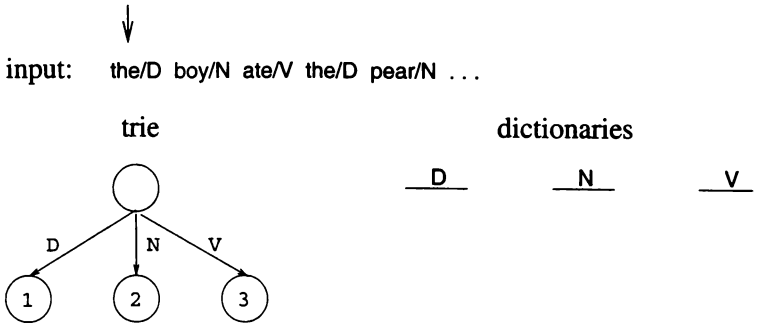
Figure 4.1 illustrates four states of the TPD model when coding the sentence "the boy ate the pear". Before any text is encoded (Figure 4.1a) all word dictionaries are empty and the trie is initialised to have one phrase for each possible part-of-speech label in the tagging scheme—in this case phrase 1 is a D for determiner, phrase 2 is an N for noun, and phrase 3 is a V for verb. The first word/tag pair is read from the input and a match is found for the tag context D leading to phrase 1. The next pair is read, but no match is found for the tag N leading out from phrase 1, therefore the code for phrase number 1 is output. The phrase has one word, which has not been seen, so "the" is coded explicitly and then saved in the corresponding dictionary for category D.

A new phrase is created by appending the unmatched tag as a leaf node to the phrase just output. In this instance, the N for "boy" is unmatched, so a new node is added below phrase 1 and assigned the lowest unused phrase number (phrase 4 in Figure 4.1b). The transition to the new phrase is labeled with the unmatched tag. Note that this phrase could not be used to encode the first two input pairs because it did not exist at the time. So it is that new phrases are created in anticipation that they may be useful at a later time.

The word associated with the first unmatched tag is coded as if it has not been seen at all—that is, as a new phrase. Starting from the root node, the longest matching tag trace is again found for the input. In this case, the tag for "boy" leads to phrase 2 (Figure 4.1c). The next pair, ate/V, is read but no match is found leading out from phrase 2 for its tag. The code for phrase number 2 is output and, as "boy" has not been seen before, the word is coded explicitly and saved in the appropriate dictionary. Again, a new phrase is created as an extension to phrase 2 and the transition to it is labeled with the unmatched tag. The process then repeats, always looking for the longest trace that begins with the last unmatched tag.

Figure 4.1d demonstrates how optimistic phrase creation becomes useful. When it comes time to code "the pear", the first tag leads to phrase 1 and

a) initial state



b) after encoding first word/tag pair

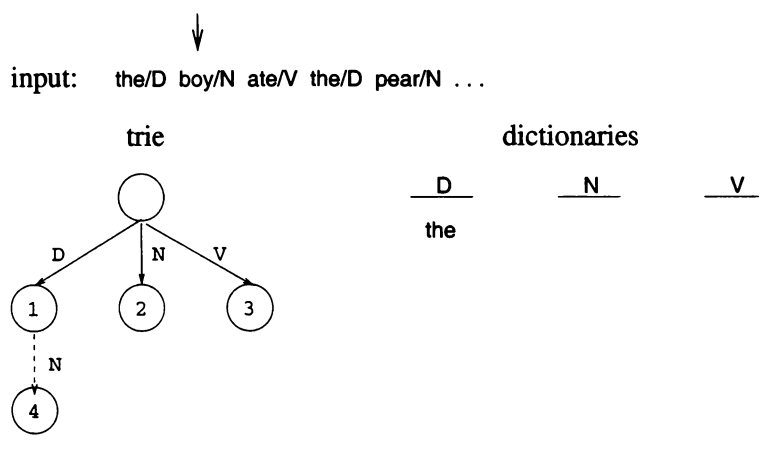
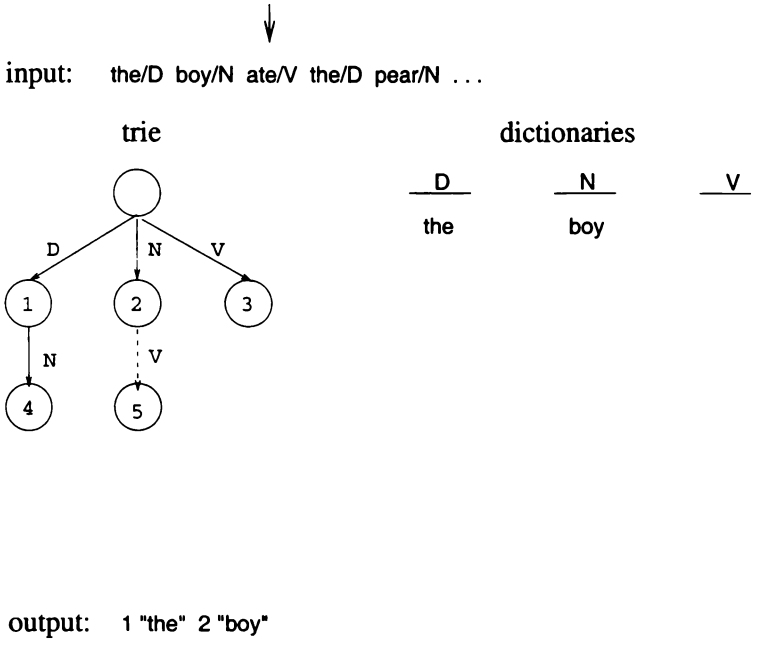


Figure 4.1: *continued next page*

c) after encoding second word/tag pair



d) after encoding fourth word/tag pair

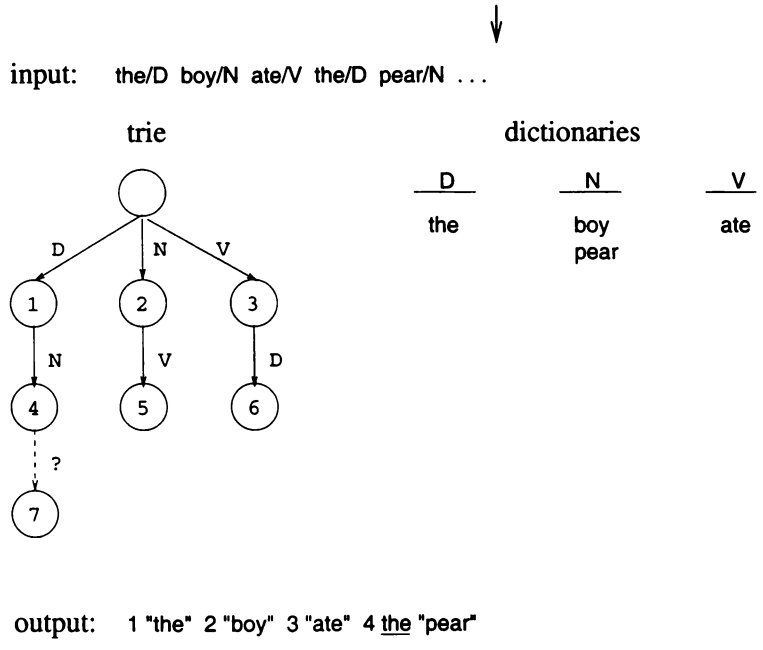


Figure 4.1: TPD states during the encoding of a tagged sentence.

the next tag leads to the as yet unused phrase 4. The next tag/word pair (whatever it might be) will not produce a match, thus the code for phrase 4 is output and a new phrase, number 7, is created ready to be labeled with the tag of the next pair. The output phrase requires that two words also be coded before the next phrase number. The first is a determiner, but one that has been seen before, allowing "the" to be encoded more efficiently with an arithmetic code. The noun "pear" is novel and so must be encoded explicitly and added to a dictionary.

The TPD decoder works by building the trie in a complementary manner. Starting with the same initial state, a new phrase is added by appending the first tag of the next phrase as a leaf node to the end of the previous phrase. Explictly coded words are added to the dictionary on first encounter, and arithmetic codes are subsequently calculated in the same manner as the encoder. Further details of the encoding and decoding processes are not relevant to this study and are therefore omitted (see [102] for a thorough treatment).

4.2.3 TPD performance

As more and more text is encoded with TPD, the tag traces stored in its trie become a better and better regular grammar approximation of its syntax (albeit through exhaustion). Given no restriction on context length, a new phrase is created after each phrase number is output, where the new phrase corresponds to the first unmatched tag appended to the longest matching trace. As a result, output phrase codes correspond to longer and longer tag sequences as the model develops. The net result is a very efficient *per tag* encoding.

Bell *et al.* [9] offer a proof to show that the coding rate of such greedy dictionary schemes asymptotically approaches the entropy of an ergodic source model—in accordance with Solomonoff's conjecture [107] that all inductive inference problems can be expressed in terms of extrapolation from a very long sequence (Section 1.2). However, in practice, character-based dictionary schemes tend to converge on source entropy more slowly than purely statistical models. In the case of TPD, this could undermine its utility for obtaining good complexity estimates for finite samples of language—even quite

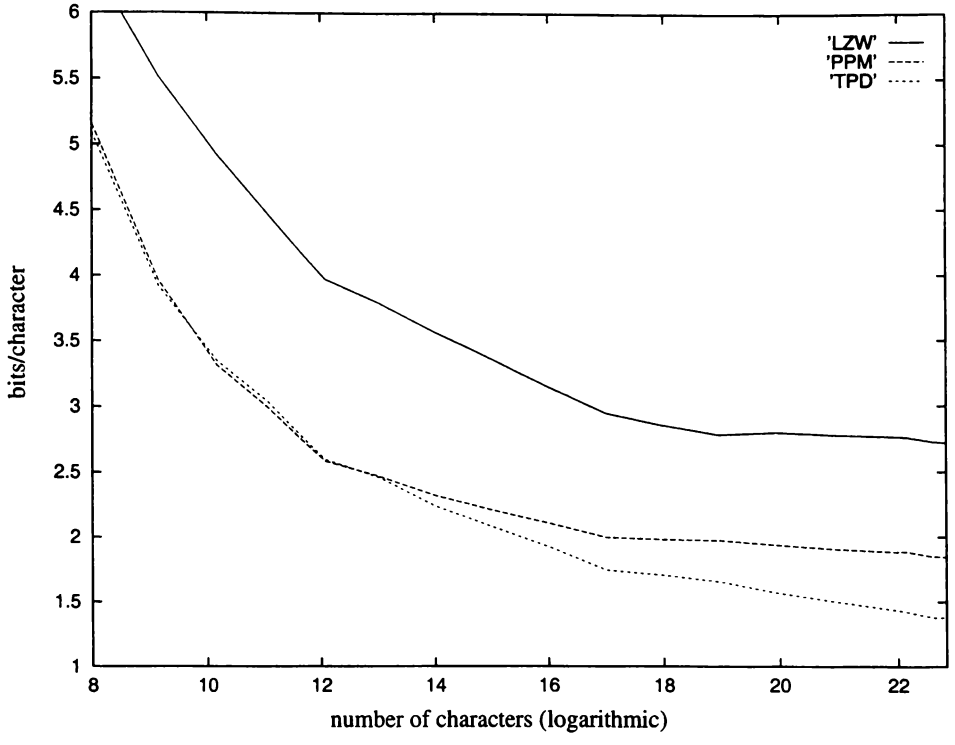


Figure 4.2: Rate of convergence for LZW, PPMC and TPD on Brown Corpus.

large ones. As it happens, the generality of the equivalence classes in the TPD model allows it to converge on source entropy as quickly as an adaptive character-based finite-context model.

Figure 4.2 plots the rate of convergence for TPD when compressing the tagged Brown Corpus. The vertical axis of the graph is the compression ratio, measured in bits per character, and the horizontal axis is the amount of text processed, expressed as the base-2 log of the number of input characters that have been compressed. Included in the graph is the corresponding compression rate for standard character-based LZW [112], along with the compression performance of a well known adaptive statistical context model, PPM [31]. This particular version of PPM is an order-3 character model using escape method C to code unseen events (see page 146 of [9] for more details of this escape method). The graph clearly shows a comparable rate of convergence between TPD and PPM up to about 10,000 characters, after which TPD starts to perform better. Whatever other conclusions may be

suggested by the graph, it is sufficient to observe that the TPD method converges sufficiently quickly to give useful complexity estimates for subsequent experiments.

4.2.4 Structural complexity

While there is empirical evidence to suggest that TPD makes good use of the structural information expressed in the part-of-speech labels of a tagged text, there may be an alternative explanation. It is possible, for example, that TPD makes all of its gains through arithmetic coding of words, and that the information content of the tags is so low that the penalty for encoding them is almost insignificant.

One way to determine whether TPD is gaining anything from syntactic regularities involving lexical classes is to change the tagging scheme in such a way as to impose a regular artificial structure on the training text. If the structure given by the original tagging scheme results in a better complexity estimate for the text than is obtained from the artificial scheme, it would suggest that TPD is successful in exploiting the structural regularity of lexical categories.

Two experiments were devised. In the first, sentences are tagged in such a way that the first word of each sentence has a unique common tag, all second words have another distinct tag, all third words another, and so on, until the end of the sentence. More formally, the set of k distinct tags used in the original tagged text is ordered to produce an analogous set $T = \{t_0, t_1, \ldots, t_{k-1}\}$, and the i-th word of each sentence is labeled with the tag $t_{i \bmod k}$; except end-of-sentence markers whose tags are unchanged and not used anywhere else. In this way, every sentence is given a highly regular structure to the extent that the first n words of any sentence have exactly the same tag structure, and each tag can therefore predict with almost perfect accuracy what tag will appear next.

In the second experiment, end-of-sentence markers and determiners are left with their original tags. All other words are tagged in sequence from T in the same manner as the first experiment, except that each time a determiner is encountered, subsequent words are tagged in sequence starting from the determiner's tag in T. That is, if the determiner tag is t_j, then the i-th word

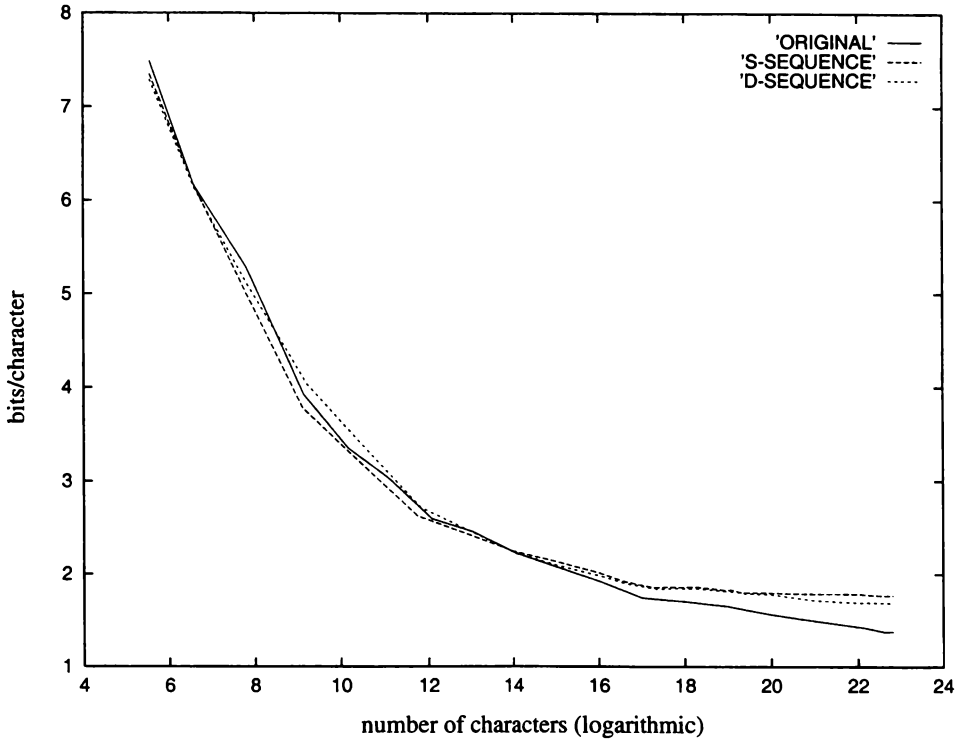


Figure 4.3: Effects of imposed structure on TPD compression rates.

after the most recent determiner is tagged with $t_{(j+i) \bmod k}$. Determiner tags and end-of-sentence tags are reserved exclusively for their original purpose (i.e. they are *skipped* in the sequencing).

Figure 4.3 plots a comparison of the progressive compression ratios when TPD is applied to the original tagged Brown Corpus (ORIGINAL), the sentence sequential structure of the first experiment (S-SEQUENCE) and the determiner sequential structure of the second experiment (D-SEQUENCE). Surprisingly, both artificial structures result in good compression. However, it is clear that the original tag scheme supplies the compressor with better structural regularity—that is, regularity more useful for translating part-of-speech information into lower complexity estimates. To some extent, one might expect the determiner sequential scheme to accidentally capture some nounphrase regularity present in the original tagged text, making its compression results closer to those for the original than for those from the simple sentence sequential scheme, and this is borne out in the evidence of the graph.

Additional conjectures might also be made, but no other conclusion need be suggested from these experiments except that TPD is indeed influenced by the accuracy of the classification scheme.

4.2.5 Tagging accuracy

That the artificial structures used in the experiments of the preceding section result in poorer entropy estimates than the original tagging scheme—despite greater structural regularity—is easily understood. Given that in both experiments words are tagged based on their position relative to either the sentence start or the most recent determiner, words end up being attributed to categories in an entirely arbitrary manner. Ignoring the fact that English sentences do in fact demonstrate some lexical patterns in their first few words, *in the limit* one would expect these contrived tagging schemes to eventually put every word into every class. Even a modestly large sample text like the Brown Corpus is likely to produce a classification where tags are of little help for predicting words.

Given that estimates of grammatical complexity are sensitive to the accuracy of the tags, this raises the question of whether there is some ideal tagging scheme for optimising the estimate. The Brown Corpus used in these experiments was tagged with the AMALGAM automatic tagging system, whose designers claim a very high level of tagging accuracy—as much as a 97% correspondence to manual tagging. But there are a good many other automatic tagging algorithms that also claim to give very accurate results [35, 71, 93]. Their outputs would certainly differ for the same input text, and it is likely that one would produce a baseline Brown Corpus text with a better TPD compression ratio than the others. But it may also be the case that the output from one of the other systems would give a better result for a different training text.

In addition to the various tagging algorithms, there is also a plethora of alternative tag sets available. Some are quite large—such as the LOB set with 153 distinct tags [60]—and discriminate between different verb tenses, noun declensions, and so forth through subcategorisation. Others—like the UNIX `parts` set of 19 tags [79]—do not make such distinctions, but label all verbs and all nouns, for example, uniformly with gross category tokens.

Given the many syntactic-agreement relations in English, such as number or person agreement between a subject noun and main verb, subcategorisation labels can help a structural model like TPD access subtle class dependencies that are obfuscated under a more crude labeling scheme.

A series of experiments was devised to determine the extent to which the refinement of a lexical classification system influences the ability of a class-based model to reduce entropy estimates of language. The goal was not so much to find an optimum classification as it was to ascertain some of the characteristics of category differentiation that are most useful to a structural model. The ultimate aim is to develop an approximate classification that is sufficient for preserving many class dependencies, but one that can be inferred easily and thereby eliminate the prior knowledge required by class-based entropy models.

4.2.6 Class complexity

Kazman [65] summarises a widely held linguistic view that much of the syntax of a language may be characterised by the inventory and properties of its functional categories (i.e. its closed-class words). He further proposes that the acquisition of syntax by a developing child is a progression from an invariant base (i.e. innate grammar) to an articulated view of functional categories. The results from word-based n-gram experiments in Chapter 2 perhaps support these conjectures in that combinations of closed-class words appear to make the greatest individual contribution to low complexity estimates of language.

An experiment was devised to examine the extent to which functional categories contribute to the structural regularity of language. Only two of the 102 most frequent words in the Brown Corpus are not function words, and those 100 most frequent function words account for more than 50% of the tokens in the text. Using these words as a simple approximation of the closed-class, the corpus was re-tagged so that the part-of-speech labels for these words were left unchanged from the original, while all other words were tagged with a common label, effectively treating them as one large thematic category. The text was then compressed using TPD.

Figure 4.4 plots (among other things) a comparison between the TPD

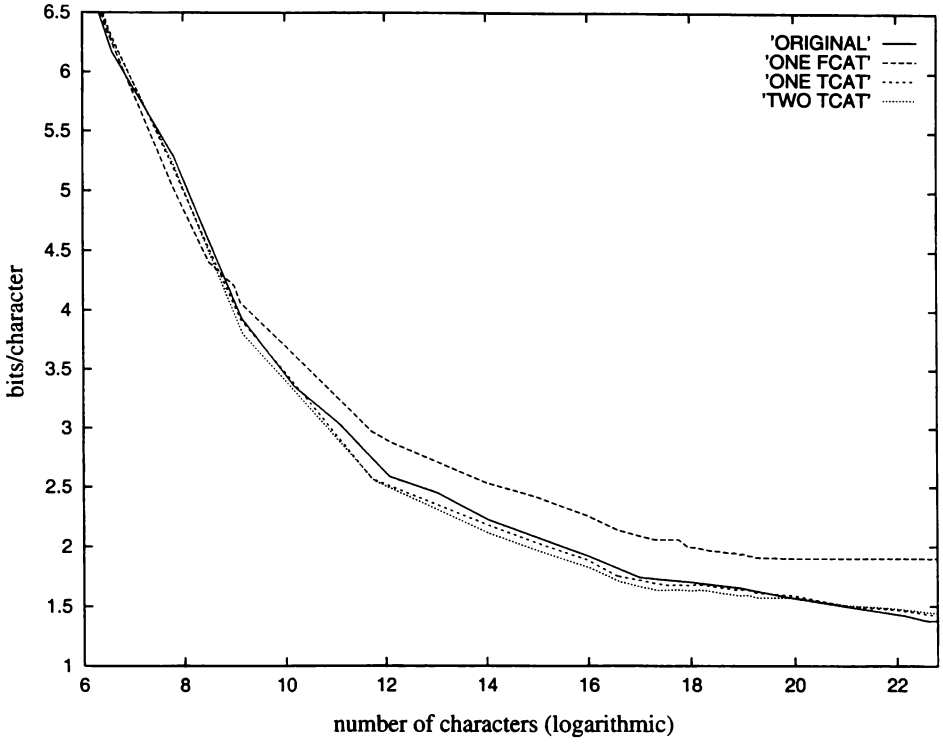


Figure 4.4: Effects on TPD compression rates of simplified FCs and TCs.

compression performance on the original tagged text (ORIGINAL) and on the altered text (ONE TCAT). The graph indicates that very little structural information is lost to TPD despite having 99.8% of its vocabulary treated as a single class. This suggests support for Kazman's conjecture—that almost all syntactic structure is captured in the juxtaposition of functional terms.

To test this further, a second experiment was devised where those 100 most frequent function words were collapsed into a single functional category and tagged with a common part-of-speech label, while the tags of the remaining vocabulary were left unchanged from the AMLAGAM output. The result after compressing this text with TPD (plotted in the graph of Figure 4.4 as ONE FCAT) was a substantial increase in the entropy of the text, apparently confirming the hypothesis that syntax is embodied primarily in closed-class words.

Given that the top 100 function words account for more than half of the text, it is possible that the increased entropy observed in the second test is

simply attributable to loss of meaningful structure for so much of the text, rather than as a symptom arising directly from differences between open- and closed-class words. A third test was therefore devised to see if increased precision in the tagging of thematic words within the parameters of the first test would result in a net gain in compression.

82% of the vocabulary of the Brown Corpus is comprised of nouns, and these constitute about a quarter of all tokens in the text. By repeating the first experiment *after* adding nouns to the set of words whose category labels are left unchanged from the output of the AMALGAM tagger, the number of words in the input that are accurately tagged is increased from just over 50% to about 75%. More plainly, the 100 most frequent function words and all nouns are labeled as in the original tagged text, and the rest of the vocabulary is treated as a single unique class. This has the effect of splitting the class of thematic terms into two subclasses: one comprised of nouns, and the other made up of all the rest (primarily verbs and adjectives).

The newly tagged text was passed through TPD and the resulting rate of compression is documented in the graph of Figure 4.4 (TWO TCAT). This experiment appears to indicate that the additional information made available when discrimination between nouns and other *non-function* words is supported does not translate into improved complexity estimates for the text as a whole. Although gains of about 0.1 bit per word over a single thematic category scheme are realised up to about one million characters, beyond this the extra precision of the noun tag offers no improvement. One possible explanation is that any decrease in the entropy of the nouns arising from the additional precision in the tagging scheme may be offset by increased entropy in the class structure due to greater detail in the contexts. Given that a thematic tag in this last experiment divides the set of thematic words from the first experiment almost exactly in two, it should improve the code for the associated word by about one bit; but it may also double the number of distinct contexts that include thematic tags, thereby increasing the length of phrase codes by one bit.

One reason why this explanation is unlikely to be correct is that it assumes rather specific characteristics for the contexts. The assumption that the number of phrases doubles when a distinction is made between noun and

non-noun thematic words rests on an additional assumption that the set of contexts involving a noun ends up being duplicated but with the noun tags replaced by non-noun tags. If this were true, then the structural roles of nouns and non-nouns would be identical and splitting the category would be pointless, as the compression results indicate. As it is generally not true that nouns and, say, verbs participate in the same grammatical structures, however, the phrase dictionary is likely to be entirely different.

A more plausible explanation is that the structural information contained in the noun tag is actually redundant. Given that determiners predict nouns very well on their own, any class n-gram that includes a determiner tag already embodies the necessary context to predict a subsequent noun tag accurately, thus the appearance of the noun tag is inconsequential with respect to the model's ability to select the specific noun in question. Other class-based contexts, such as those with prepositions, may also contain the necessary information to predict nouns accurately without the need for noun tags, and similar circumstances may exist for other thematic words as well—as when auxiliaries predict verbs.

Understanding why subcategorisation of non-function words does not appear to improve structural predictions is not expedient to the purposes of this thesis. It is sufficient to observe that almost all category structure required for obtaining good complexity estimates from class-based n-grams is adequately represented when a distinction is made between different functional categories, and when content words are recognised as something else entirely.

4.3 Discussion

The class-based context models outlined in this chapter attempt to make gains on the basis of two structural assumptions. The first is that knowledge of a word's category increases a model's ability to predict the word itself. Section 4.1 shows that this assumption alone does not lead to improved complexity estimates because the entropy of the category information is exactly equal to the reduction in the entropy of the word it is being used to predict. To make gains from class information, the probability of a category label

must itself be improved.

The second structural assumption is that the probability of a category symbol depends solely on the context of preceding category labels. For example, the fact that determiners signal onset of a nounphrase means that appearance of a determiner label not only provides a conditional probability boost for the associated determiner but for the ensuing noun (or perhaps adjective) as well.

In principle, part-of-speech traces cannot improve lexical prediction indefinitely. In the limit, any useful context they provide will be embodied in the explicit word sequences with which they are associated. But, given that all language samples are finite, part-of-speech contexts can yield gains in entropy estimates because they generalise very quickly about very real lexical dependencies—dependencies that are most practically regarded as class-based. That is, each tag context corresponds to an equivalence class for a potentitally infinite set of word sequences (assuming new words can be introduced to a language without limit). For example, because the word "goat" can almost always be used in the same structural context as "cow", a class-based model that has seen "goat" can impart its statistics to "cow", provided it knows they belong to the same category.

The most important observation made in this chapter is that the precision of class assignment does not appear to be tremendously important for exploiting much of the benefit of class-based prediction. The experiments of Section 4.2 indicate that the distinction between open- and closed-class words is significant, as are subcategorisations of the latter. However, differentiation between subclasses of open-class words is much less important, and perhaps even entirely unnecessary for class-based n-gram modeling.

An equally important observation is that even a very crude approximation of open- and closed-class categories is sufficient to preserve the class-dependencies that are useful for category-based n-grams. The experiments outlined in this chapter more or less fixed a boundary between these two broad categories solely on the basis of a statistical heuristic—namely, the top 100 most frequent words are function words and the rest are not. The resulting compression from an unrestricted category context model was almost identical for a text tagged using this simple classification method and the

same text tagged by the more discriminate AMALGAM technique. This suggests that access to the predictive capacity of class contexts can be achieved without the *a priori* need for an accurate tagger.

Taken together, these observations create a framework for building a simple entropy-based n-gram model that exploits the regular structure exhibited by lexical categories. The goal now is to combine this with the high utility bigrams of Chapter 2 and the semantic dependencies between close proximity content words discussed in Chapter 3. This is the basis of the super-adjacency model introduced in the next chapter.

Chapter 5

Super-Adjacency Models

Three key observations are made in the preceding chapters. The experiments in Chapter 2 indicate that word-based n-grams work well for modeling adjacent closed-class words, but less well for lexical patterns involving open-class words. Chapter 3 provides evidence to suggest that a substantial amount of the mutual information in pairs of close proximity open-class words can be accessed by using the most recent open-class word as the conditioning context for predicting the next one, ignoring any intervening closed-class words. And Chapter 4 shows that the distinction between open- and closed-class categories, combined with some discrimination among closed-class subcategories, is sufficient for capitalising on structural dependencies in a class-based entropy-model. This chapter shows how all three of these properties can be exploited through a simple, but unconventional, word-based bigram technique—the super-adjacency model.

5.1 Function/content n-gram models

Linguists have long recognised a grammatical distinction between function words (also called grammatical words, closed-class words, or functional categories) and content words (also called semantic words, open-class words, or thematic categories), and there is substantial psycholinguistic evidence to suggest that these two broad lexical classes are subject to entirely different cognitive processes [48, 80, 36, 54, 50]. This section outlines early attempts to exploit the differences between these two lexical types through unconven-

tional n-gram models of language. While such systems have been able to achieve some level of satisfactory performance, they have not been able to do so without interpolating the statistics of conventional word-based n-grams into the probability estimates. However, an analysis of their methods and results suggests that a much simpler approach might be more successful.

5.1.1 A particle/content bigram model

Isotani and Sagayama [56] observed that Japanese sentences have a very regular form when a distinction is made between content words and particles. Specifically, a Japanese sentence consists of phrases, each of which typically consists of a content word followed by an optional particle. Thus a sentence can be regarded as a sequence of n phrases, where w_i^h is the content word head of the i-th phrase and w_i^t is the particle tail of that phrase. Isotani and Sagayama developed an interpolated bigram model that attempts to exploit this view of Japanese phrase structure as a means for achieving improved accuracy in sentence recognition tasks.

Isotani and Sagayama contend that the sequence of content words that results when all particles are ignored "is expected to statistically describe the semantic relationship between words in the sentence." Specifically, they argue that high mutual information can be presumed to exist for pairs of consecutive content words, and that this can be exploited by an independent bigram model of the content word sequence. They estimate the probability of the content word sequence formally as

$$\prod_{i=1}^{n} \Pr[w_i^h | w_{i-1}^h].$$

In a complementary fashion, they contend that syntactic constraints are exposed in the sequence of particles that results when all content words are ignored. Japanese particles serve primarily to assign case to the content words they follow. The nominative case marker *-ga*, for example, identifies the subject noun of a sentence, and as a consequence it is very rare for it to appear in successive phrases. Further, postpositions (which are also treated as particles in the model) exhibit regular sequential behaviour, as

when *kara*, meaning "from", is often followed by *made*, meaning "to". Syntactic regularities such as these can be statistically modeled by dedicated particle-to-particle bigrams, and the probability of the particle sequence can be calculated in isolation using

$$\prod_{i=1}^{n} \Pr[w_i^t | w_{i-1}^t].$$

If this view of Japanese sentence structure was sufficient to cover all well-formed expressions, and if independence between the particle and content word sequences can be assumed, then sentence probabilities could be estimated as

$$\Pr[S] \simeq \prod_{i=1}^{n} \Pr[w_i^t | w_{i-1}^t] \times \Pr[w_i^h | w_{i-1}^h].$$

Unfortunately, there are a number of sequential exceptions that upset this calculation. One difficulty is that the first phrase of a sentence has no prior context for predicting its particle or content word. To avoid this, Isotani and Sagayama assume a dummy phrase[1] at the start of each sentence. A similar problem arises when a content word is not accompanied by a particle—as with nouns whose case is unmarked, such as ergative or absolutive. In situations such as this, their model assumes a *null* particle between the adjacent content words.

The situation is more troublesome for adjacent particles, as when several auxiliary verb particles are applied to the main verb. In such cases, Isotani and Sagayama choose simply to ignore all but the last particle. While at first it may seem an unsatisfactory feature of a language model to simply ignore terms in this way, it is important to remember that Isotani and Sagayama were not interested in complexity measures *per se*. Their goal was instead to develop a solution to the candidate sentence selection problem [40] and, to that end, they needed only the ability to assign a higher probability to a correct response than to any other candidate sentence.

[1]The exact nature of the dummy phrase is not elaborated upon in their report [56].

5.1.2 Experiments with particle/content bigrams

To examine the extent to which particle-to-particle bigrams (hereafter called P–P bigrams) and content word-to-content word bigrams (hereafter called C–C bigrams) can contribute to accurate candidate selection, Isotani and Sagayama conducted a series of tests with twelve samples of spoken Japanese sentences. They compared the accuracy obtained using just P–P bigrams against that obtained from using just C–C bigrams, and also against a third model that combined the two (hereafter called the P–P + C–C model). While the two simpler models achieved comparable levels of accuracy, the particle bigrams fared slightly better overall—with an average of 69.2% for the P–P model as compared to 68.3% for the C–C model—and outperformed the content word model in eight of the twelve individual tests. In all cases, however, the combined P–P + C–C model outperformed both of the dedicated bigram models; by at best 6.9% and at worst by just 0.3%. The researchers concluded that P–P bigrams and C–C bigrams did in fact capture independent linguistic information.

Isotani and Sagayama hypothesised that strong inter-phrase correlations might exist between the particle of a phrase and the content word of the next phrase—dependencies that could not be captured in the P–P + C–C model. To test this, they constructed a model based just on particle-to-content word bigrams (hereafter called P–C bigrams) and tested it using the same twelve language samples. For seven of the samples, this model did better than either of the P–P or C–C models, and it was never worse than both. A more important observation, however, is that the P–C model did better than the P–P + C–C model a third of the time—apparently confirming the hypothesis.

To incorporate inter-phrase dependencies, Isotani and Sagayama constructed yet another model that interpolated the statistics from all three types of simple bigrams (P–P, C–C and P–C bigrams). In this model, the probability of a sentence is estimated using

$$\Pr[S] \simeq \prod_{i=1}^{n} \left(\frac{\Pr[w_i^t | w_{i-1}^t] \times \Pr[w_i^h | w_{i-1}^h] \times \Pr[w_i^h | w_{i-1}^t]}{\Pr[w_i^h]} \right).$$

Subsequent experiments found the overall average performance of this model to be better than the P–P + C–C model (73.8% compared to 72.4%), but

worse in four of the twelve individual tests.

On the basis of all their experimental results, the researchers concluded that the three basic bigrams (the P–P bigrams, C–C bigrams and P–C bigrams) "have independent information about inter-phrase connectivity", and that an interpolated model that utilises all three bigram types simultaneously "can catch the syntactic and semantic relationships between words in Japanese sentences by estimating probabilities from a large text database."

5.1.3 Analysis of particle/content bigrams

Isotani and Sagayama do not include specific sentence probabilities in their report, and in the absence of this information it is difficult to say how effective their approach might be as an entropy-based model of language. Even comparisons against conventional bigram models would provide a basis for speculation, but these also do not appear as part of their experimental results. Nevertheless, there are observations and issues related to their studies that merit some comment.

One can view the set of terms constituting the vocabulary of a language as being the union of two non-overlapping subsets: a subset F of all functional terms and a subset T of all thematic terms (i.e. content words). A conventional bigram model of the language thus has $(F+T)^2 = F^2 + T^2 + 2FT$ independent parameters. The interpolated model proposed by Isotani and Sagayama employs three different types of bigrams in its calculations. The P–P bigrams are constructed from functional terms only, and there are F^2 many of them. There are also T^2 many C–C bigrams, and $F \times T$ many P–C bigrams. Their interpolated model thus has $F \times T$ fewer parameters than the conventional model, where the missing elements are of course the intra-phrasal C–P bigrams.

In light of the claim by Isotani and Sagayama that "particles mainly convey case information about the content words preceding them", one might expect a strong correlation between the content word and particle within a given phrase, and that exclusion of C–P bigram statistics would therefore undermine the accuracy of their probability estimates. The researchers explain that their decision to omit these bigrams is based on the fact that input sentences are parsed into phrases prior to training, and that "an intra-phrase

CFG is used for phrase recognition, therefore, only the particle-to-content word bigram is taken into account". The intra-phrase information to which they allude is specifically that of content-to-particle dependencies.

This explanation does not obviate the question as to whether estimates of sentence probabilities can be as good as those given by a conventional bigram model if C–P bigrams are not factored into the calculation. Given that the three bigram types they ultimately use all target inter-phrasal dependencies, these could not subsume the missing information. In order for this method to be more effective than the conventional model, then, the predictive contexts discovered through syntactic abstraction must be sufficient to compensate for any loss attributable to the unused intra-phrasal bigrams. Obviously P–C bigrams cannot make up this deficit as they correspond to conventional word bigrams, thus the gains would have to come from either or both of the P–P or C–C bigrams.

Languages that have overt case-marking, as Japanese does, generally permit more liberal word-order in sentence construction than is possible for uncased languages, which must rely on fixed canonical orders to imply case, as with the subject-verb-object order of English. Japanese does have a preferred canonical word-order of subject-object-verb, but a speaker can move any nounphrase to the start of a sentence if it is desirable to increase its importance in the utterance. In addition, some nounphrases can in many instances be assumed, and as a consequence they are often omitted from sentences. Given such flexibility in the use of case-marked nouns, it is not likely that the statistics of P–P bigrams are on their own sufficient to offset the loss of intra-phrase correlations. More plainly, a noun in a conventional Japanese bigram provides about as good a predictive context for its case-marker as would the particle of the preceding phrase. So it must be that if the model proposed by Isotani and Sagayama is in any way superior to conventional bigrams it is because of its ability to better exploit the mutual information of close proximity content words through C–C bigrams.

5.1.4 A function/content word trigram model

Like Isotani and Sagayama, Geutner [46] observes that standard word-based n-grams generally fail to capture genuine linguistic constraints. As an alter-

native, she proposes a model of English that uses the distinction between function words and contents words to gain access to the local constraints that effectively correspond to syntactic and semantic knowledge.

Geutner's idea is to use an unconventional trigram model, where the prediction of the word w_i is based on the context of the preceding word w_{i-1} combined with either the most recent content word **c** if w_{i-1} is a function word, or with the most recent function word **f** if w_{i-1} is a content word. More formally, the probability of a word is estimated as

$$\Pr[w_i] = \begin{cases} \text{if } w_{i-1} \text{ is a content word} & \Pr[w_i|\mathbf{f}, w_{i-1}] \\ \text{if } w_{i-1} \text{ is a function word} & \Pr[w_i|\mathbf{c}, w_{i-1}] \end{cases}$$

Geutner uses the sentence "we will ride on the bus" as an example in support of her reasoning. In a conventional model, "on the" provides the context for predicting "bus", but Geutner contends that "ride the" is just as good a predictor. Though she does not elaborate further, one might presume her claim rests on the observation that both models include the determiner, which serves to predict an ensuing noun, and while the preposition within the standard context further constrains that noun to be likely a referent to something that can have something else "on" it, the verb in Geutner's so-called function/content trigram provides a more restrictive semantic clue: that the referent will likely be something that one can "ride" and, hence, implicitly be "on" as well.

Despite the apparent soundness of her function/content word trigram idea, empirical studies by Geutner produced much higher complexity estimates than those obtained by a conventional trigram model. Geutner is not surprised by this, stating candidly that "as to be expected, the function/content word model has no use on its own". She does, however, maintain that the model is able to capture distinct linguistic knowledge that can be interpolated with a parallel word trigram model to give improved complexity estimates—just as Isotani and Sagayama interpolated particle-to-content word bigram statistics in their probability estimates. Though Geutner does not include a description as to how the interpolation is effected, she does provide results from several experiments that show her combined model outperforming conventional trigrams by about 4%.

5.1.5 Analysis of function/content trigrams

Geutner's results are disappointing in that the function/content word model does not by itself make good on its potential for translating syntactic and semantic dependencies into improved complexity estimates. More plainly, that the model has to be combined with standard word-based trigrams before delivering its picayune gains suggests that it fails to capture the targeted dependencies effectively.

Indeed, there does not appear to be much linguistic basis to Geutner's approach. Consider that there are only four types of function/content word trigrams possible in her model.

w_{i-2}	w_{i-1}	w_1
f	c	f
f	c	c
c	f	f
c	f	c

If the word to be predicted is a content word then it is unlikely the most recent preceding function word can contribute much to conditioning an accurate probability. The function word may offer some clue about the category of the next word, as when a determiner signals imminent onset of a noun, and may even project some agreement constraint, as when a singular determiner normally constrains the following noun to be singular. But adjectives also predict nouns; so too do verbs to some extent; and a subject noun not only foreshadows the main verb implicitly but often also projects exactly the same agreement relation as might be dictated by the noun's determiner. More plainly, the most recent content word will often subsume the predictive potential of the most recent function word. Given that all contexts for predicting a content word also include the next most recent content word, the potential contribution of the function word is quite likely marginalised.

If, on the other hand, the word to be predicted is a function word then the most recent content word is similarly not likely to be of much help. A subject noun, for instance, may tend to be followed by an auxiliary, and the auxiliary may even have to agree with the noun in terms of person or number features, but the noun's determiner will often embody the very same constraint information. Further, verbs are often followed by adjunct phrases

that begin with a preposition followed immediately by a determiner. In that respect, the verb may be said to predict the determiner, but no more so than the intervening preposition. In the sentence supplied by Geutner, for example, the probability of "the" is estimated from a function/content word trigram as Pr[the|ride, on]. Though the verb transitivity certainly suggests a direct object and therefore the determiner of the object's nounphrase, the preposition "on" subsumes this predictive potential almost entirely.

Insofar as these conjectures are correct, they suggest that there is little to be gained by combining function and content words within the same predictive contexts. Any potential the lexical distinction might have for capturing syntactic and semantic dependencies would be just as accessible in a bigram model that predicts each function or content word from the most recent word of the same class, as was originally proposed by Isotani and Sagayama. What is uncertain is whether such a model can be extended as a complete account of language—without the need for dropping words or introducing dummy terms—and whether inter-class dependencies (between function and content words) can be incorporated efficiently and effectively.

5.2 The super adjacency model

If one divides the vocabulary of a language into the two broad classes of content words and function words, then language can be viewed as the interlacing of two streams, one comprised only of the content words and the other solely of function words. Two words are said to be "super-adjacent" if they are next to each other in either of the two streams. By garnering separate bigram statistics for each stream of super-adjacent terms, the unique sequential characteristics of the streams can be modeled independently. It is argued that the nature of the interaction between the two streams is implicit and class-based, not explicit and word-based, and thus inter-sequence dependencies can be captured through the use of an escape symbol that manages the transfer from one super-adjacency model to the other. The net effect is a parallel model that preserves the most useful bigrams of a conventional approach and simultaneously increases the incidence of adjacency for pairs of content words with high mutual information. The result is a more compact

model that delivers better estimates of language complexity than is possible from standard bigrams.

5.2.1 Sequential agreement effects

The ways in which words can be combined together to create well-formed sentences are governed by the syntactic constraints of a language. Such constraints may be dictated by rules that explicitly define permissible structures, or by more sublime heuristics that indicate preferential orderings for successful communication. Whatever the exact nature of the underlying generative mechanism, the output effect is a lexical sequence with conspicuous statistical properties wherever true linguistic dependencies arise.

Broadly speaking, there are three kinds of agreement phenomena apparent in the surface structure of language: 1) syntactic agreement structure emerges as intra-phrasal patterns within constituent elements, and as inter-phrasal patterns within complete sentences; 2) semantic agreement is observed as patterns of thematically related words whose juxtaposition reflects coherent intension with respect to the present topic of discourse; and 3) inflectional agreement morphology is manifest as correlations between sub-categorisations embodying more abstract lexical features like number and tense.

Syntactic agreement structures are components of isolated sentences and do not extend across sentence boundaries. Following from the arguments of Chapter 2 and Chapter 4, the essential characteristics of such structures are determined by the functional categories of a language, a view that is consistent with modern theories of syntax [44, 109, 64]. Isotani and Sagayama proposed that the statistical properties of these structures can be suitably approximated by a bigram model of the sequence of function words that remains when all content words in a language sample are ignored, and this is the approach adopted by the super-adjacency model described below. Unlike the model defined by Isotani and Sagayama, however, the entire function word sequence is modeled without alteration—that is, no functional terms are dropped or inserted to compensate for unwanted irregularities.

Semantic agreement pertains to the appropriate combination of thematic terms to produce coherence of meaning in sentences. In broader discourse,

semantic regularities may extend across sentence boundaries given that intension of adjacent sentences are often related. Following the argument of Chapter 3, semantic agreement is reflected in high mutual information for pairs of co-occurring words, and it can to a great extent be captured by a bigram model of the content word sequence that results when all function words are ignored. This is the view taken by Isotani and Sagayama in their particle/content word model, and it is maintained here in the super-adjacency model. However, unlike the particle/content word model, super-adjacency ignores sentence boundaries in the content word sequence and therefore has no need of dummy content words to provide context for the first semantic term in a sentence.

Inflectional agreement morphology is the extension to structural agreement needed to enforce compatibility for the inflectional attributes of words—as when a noun must agree in number with its associated determiner (e.g. "that dog" as compared to "those dogs"). Manifestation of this form of agreement often involves morphological transformations that modify a base semantic root to an appropriately inflected surface form, frequently by affixing a inflectional morpheme—such as the *-ed* or *-ing* tense suffixes of regular English verbs, or the *-s* plural marker for countable nouns. The statistical effect of inflectional agreement is strong correlation either between inflection markers or between inflection markers and function words. In fact, it is generally held that inflectional affixes are distinct elements within the functional categories of a language [48, 66], and their agreement effects are more properly viewed simply as regular syntactic agreement structures [64, 29]. If inflection markers can be disentangled from their associated roots and treated as independent function words, then their agreement behaviour is subsumed by a model of syntactic agreement structures. This is the view taken by the inflectional super-adjacency model outlined in Chapter 6. The remainder of this chapter, however, is restricted to the more fundamental problem of modeling whole-word agreement phenomena.

5.2.2 A bigram model of super-adjacency streams

Given the above descriptions of the particle/content word bigram model by Isotani and Sagayama, and the function/content word trigram model by

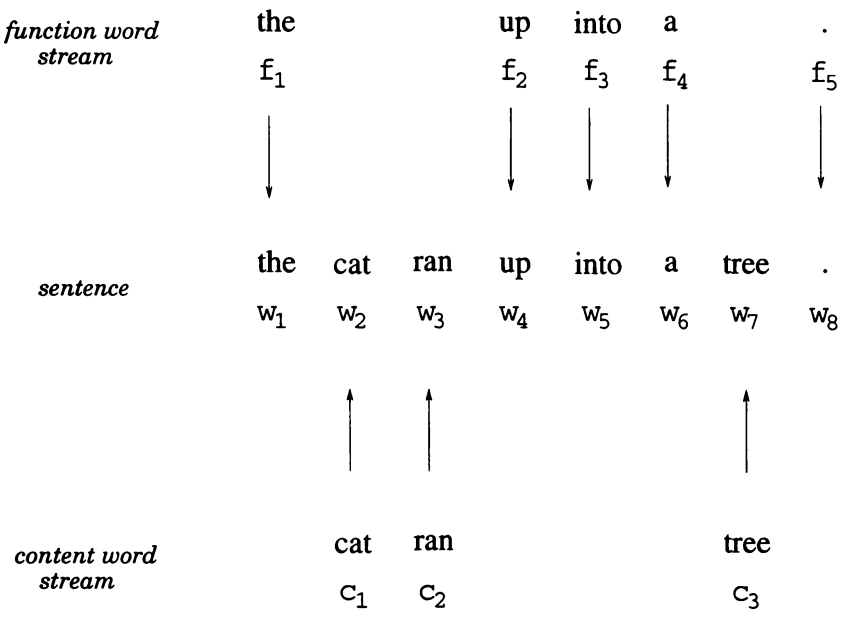


Figure 5.1: A sentence as interlaced function and content word streams.

Geutner, the basic mechanism of a super-adjacency model is easily stated. A language sample is viewed as the product of interlacing two separate lexical streams: one stream comprised entirely of content words and the other entirely of function words, as depicted in Figure 5.1. The probability of a word w is made conditional on the context given by the most recent word of the same type. Formally, if $\mathbf{c}$ is the most recent content word and $\mathbf{f}$ the most recent function word, then

$$\Pr[w] = \begin{cases} \text{if } w \text{ is a content word} & \Pr[w|\mathbf{c}] \\ \text{if } w \text{ is a function word} & \Pr[w|\mathbf{f}] \end{cases}$$

End-of-sentence markers are regarded as function words by the super-adjacency model, thus the first function word of a sentence has the end-of-sentence marker from the previous sentence as its conditioning context. In comparison, no sentence boundaries exist in the stream of content words, and the first content word of a sentence is predicted based upon the context of the last content word in the preceding sentence.

The probability of a sentence can, in part, be calculated as the product of the conditional probabilities of its words. Given a sentence with n content words and m function words, where $\mathbf{c}_0$ is the last content word in the preceding sentence and $\mathbf{f}_0$ is the end-of-sentence marker for the preceding sentence, the probability of the sentence S produced when the two streams are combined can be approximated as

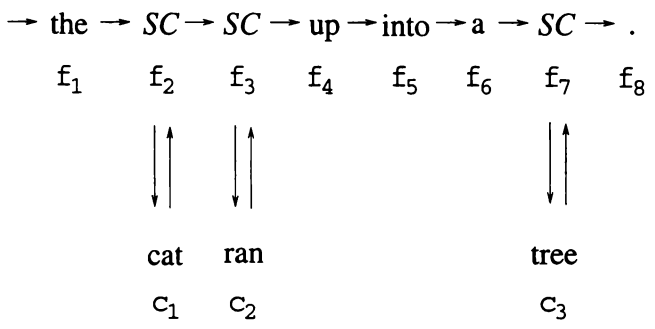
$$\Pr[S] \simeq \prod_{i=1}^{n} \Pr[\mathbf{c}_i|\mathbf{c}_{i-1}] \times \prod_{j=1}^{m} \Pr[\mathbf{f}_j|\mathbf{f}_{j-1}]$$

This formula does not by itself assign a probability to a specific sentence; rather it assigns a probability to the set of sentences comprised of the specified function and content word sequences. No account is provided for the precise manner in which the two streams interlace. Viewed another way, given the set of bigrams used to estimate the probability from the formula above, it is possible to reconstruct the sentence only to a limited extent. The function words involved, and their order with respect to each other, are known, as are the content words and their order, but the exact points where the two streams come together is unspecified. Given the stream of function words and the stream of content words, it is possible for a speaker of that language to make quite sound guesses as to how the two are interlaced, but the additional linguistic and world knowledge required to do so is not available to a sequential model. The uncertainty of the interaction necessarily entails additional cost in the entropy of the sentence, thus the goal is to find a way to model it as efficiently as possible.

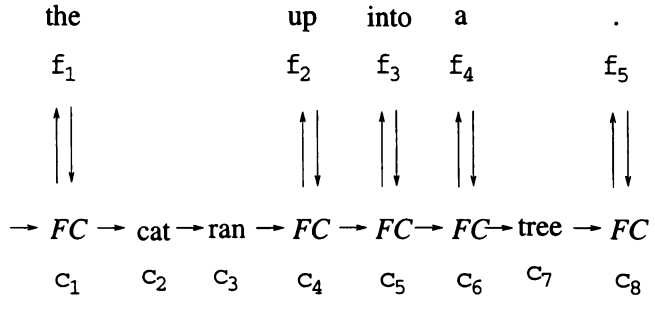
5.2.3 Modeling stream interaction

There are three methods by which an n-gram model can provide an account of how the streams of function words and content words interlace, and illustrations of these methods are given in Figure 5.2. One way is to insert a marker (SC) into the stream of function words at each point where a content word ought to appear. The marker is treated as an additional functional term: a category symbol for the set of semantic terms. The modified functional stream is modeled independently using function word bigrams; but

a)



b)



c)

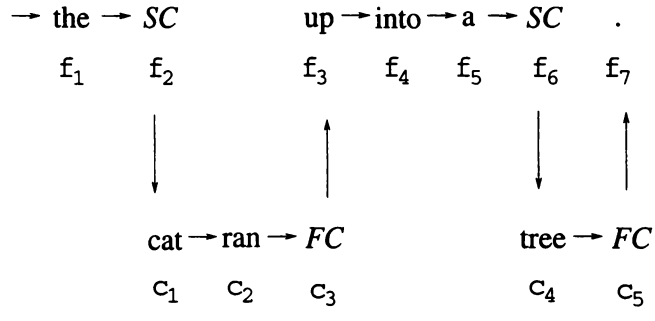


Figure 5.2: Three options for modeling stream interaction.

whenever a semantic category symbol is encountered, processing is momentarily transferred to the content word bigram model. The content word is predicted and processing returns to the function word model immediately afterward.

A second method, depicted in Figure 5.2b, is the complement of the first.

A marker (FC) is inserted into the stream of content words wherever a function word would normally appear. The marker is treated as an extra content term: a category symbol for the functional class. The content word stream is modeled using content word bigrams and processing is momentarily transferred to the function word bigrams whenever a function category symbol is encountered. After the function word has been accounted for, processing returns to the content word model.

The third option, illustrated in Figure 5.2c, is a blend of the first two. A semantic category symbol is inserted into the function word stream at each point where a contiguous sequence of content words would normally be found. Similarly, a functional category symbol is substituted into the content word stream for each contiguous subsequence of function words. The functional stream is modeled by function word bigrams and when a semantic category symbol is encountered processor control is transferred to the content word model. The content word model continues to be used until a functional category symbol is discovered, at which point control is returned to the function word model, and so on.

There are a number of reasons why the first method is preferred. First, the position of content words is a property of syntax, and syntactic dependencies are largely given by the functional structure of sentences. Moreover, function words often foreshadow a subsequent content word at the category level—for example, determiners predict adjectives and nouns quite well, and auxiliaries predict verbs. Content words, on the other hand, do less well at predicting functional terms. While a subject noun may, for example, increase the probability of, say, an auxiliary, there is nothing in an n-gram model to provide the necessary case information for this. A noun is just a noun and as such offers little clue as to whether the next word will be a verb, an auxiliary, or an end-of-sentence marker.

Second, both of the other methods involve interrupting the stream of content words with a functional category symbol. Having category symbols adjacent to content words interferes with the content word bigram model's ability to exploit the mutual information of close proximity semantic words. This could be avoided by ignoring functional category symbols when it comes time to predict the next content word, so that when control returns to the

semantic model the next content word is predicted based upon the context of the most recent content word regardless of whether there is an intervening functional category symbol. Unfortunately, this increases the statistics of the predicting context as it must now predict both the adjacent functional category symbol and the content word that follows. To avoid this, an additional set of content word bigrams must be maintained for situations where a category symbol must be ignored, but this entails a significant increase in the size and complexity of the model.

Regardless of speculations about what may happen, the most important reason for using the first method is that it gives better entropy results in empirical studies. Before the relevant experimentation can be discussed, however, it is necessary to clarify one more detail of the super-adjacency model: how the distinction between function and content words is made.

5.2.4 The function word class

Fundamental to the super-adjacency model is a requisite ability to distinguish between function words and content words. Given that these two broad categories are nonoverlapping in the super-adjacency model, it is sufficient to define just one of them; the other is implied as the complement portion of the vocabulary. The set of content words is often referred to as the *open class* because it is subject to limitless new additions. For example, new proper nouns for people, places and products, the neologisms of technological jargon, and words borrowed from foreign languages are constantly incorporated into the vocabularies of the world's languages. In comparison, the set of function words is often referred to as the *closed class* because its membership is more or less fixed. That the number of function words is very much smaller than the number of content words is a further incentive to define the closed class.

As far as intuitions go, linguists are more or less agreed about the properties that distinguish a function word from a content word. Function words are generally more abstract in meaning than content words, with their role in language being primarily syntactic as opposed to semantic. They typically do not bear stress in spoken language, though deliberate emphasis is possible, and they do not enter freely into word formation processes, like compounding and derivation. Furthermore, the fact that function words tend to appear late

in the productive vocabulary of children acquiring their first language, and that the facility for recognising function words can be selectively impaired in aphasic adults suggests they are subject to different mental representations and processes than content words [20].

Despite general agreement about the linguistic attributes of function words, there is no clear consensus as to precisely which words are functional and which are not. Even so there are a number of heuristics available for approximating the closed class. Most are based on a conspicuous statistical property of function words; namely, that they are generally very much more frequent than content words.

Dictionary method

Function words are exemplified by minor grammatical categories, including prepositions, determiners, auxiliaries, pronouns, possessive adjectives, and so forth. Viewed another way, content words are nouns, verbs, adjectives and adverbs, and function words are pretty much all the rest. Thus one way to derive the set of function words is simply to scan through a dictionary and gather up all words that do not belong to a content word category.

There is some difficulty with this method in that a few words belong to both semantic and functional categories (such as copula verbs, some interjections and some prepositions). A conservative *ad hoc* solution might be to only include a word if none of its listed categories are of the four major semantic types. But, given that the ultimate goal of the language model is to deliver low entropy estimates, it is perhaps undesirable to exclude a high frequency word that is most often used as a grammatical term just because it may occasionally be used in an overtly semantic sense. Thus a more practical heuristic might be to make decisions based upon the preferred (i.e. first) category associated with a word. Using a machine readable version of the Oxford English Dictionary, 248 words of the Brown Corpus vocabulary are deemed to be members of the closed class using this selection criterion.

Top 100

The dictionary method outlined above differentiates between function and content words based upon the grammatical judgments of language experts—

specifically, judgments made by the lexicographers who compiled the dictionary. While linguists may disagree as to the which words are functional and which are not, the judgments made by the lexicographers are perhaps as sound as can be hoped for. In the absence of expert linguistic knowledge, however, decisions about closed-class membership can still be made based on the statistical properties of function words.

Caplan claims that "there are approximately 500 or so function words in English, and, of the 100 most common words in English, most are function words" ([20], page 267). On the assumption that the grammatical importance of the other 400 function words diminishes proportionately with their decreasing frequency, a practical approximation of the closed class might simply be to use the 100 most frequent words from a large representative sample of English. While some erroneous inclusions are likely to occur from this method, and some oversights certain, the statistical significance of the 100 most frequent words may be sufficient for the purposes of a super-adjacency model.

1% solution

The errors and omissions that occur when the 100 most frequent words are defined to be the closed class arise in response to specific characteristics of the sample used. If the training text happens to include excessive treatment of, say, household appliances, there is some risk that words like "toaster" and "refrigerator" will make it into the approximated closed class, even though they are not typically among the 100 most frequent words of English in general. Moreover, some *bona fide* function words may just fall outside of the top 100 as a direct consequence of two or three content words appearing with unusual frequency in the training sample.

Side-effects from inaccurate lexical probabilities attributable to the idiosyncrasies of the sample can be mitigated by taking the intersection of vocabularies from several large and disparate language sources before garnering the set of function words. Because function words are grammatical terms required by English syntax as a whole, they have a high likelihood of appearing in *any* sufficiently large sample, and thus will be preserved when the vocabularies of several such samples are intersected. Conversely, while

any one sample may have some number of unusually frequent semantic terms, it is unlikely that such terms will occur in all large samples, thus they are eliminated when the intersection is taken.

Once idiosyncratic terms are removed, the proportion of function words at or near the top of the frequency-sorted intersection vocabulary tends to be greater than it is in the vocabulary of any one sample. This permits consideration of a greater number of *most frequent* words when approximating the closed class. Given the ratio of function to content words in English, it has been proposed that the top 1% of most frequent words in an intersection vocabulary is a good approximation [104]. When this approach is taken using machine readable versions of four large novels, each with vocabularies greater than 11,000 words—*Wuthering Heights*, *Moby Dick*, *Far from the Madding Crowd*, and *Dracula*—the result is a closed class with 83 words.

Split points

Two other heuristics have been proposed to partition a frequency-sorted vocabulary into the sets of function words and content words. One is based on the observation that approximately half of all tokens in a typical sample of English are function words [98]. The suggestion is that a fair approximation of the closed class can be established using the minimum number of most frequent words needed to account for 50% of the tokens in a large representative sample of English. Using the statistics of the Brown Corpus, 102 words are included in the functional set using this method—just slightly more than the top 100.

Another technique was proposed to isolate content words for a model that automatically derives inflectional suffixes [100]. The idea is based on Zipf's *principle of least effort*—a theory advanced to explain the many harmonic distributions found in a variety of data sets, but a theory that is perhaps more impressive than perspicuous [115]. The partitioning algorithm plots the hyperbolic distribution of a vocabulary ordered by frequency, then divides it into function words and content words at the knee[2] of the curve. Applied to the vocabulary of the Brown Corpus, the 210 most frequent words are established as the closed class. Subjective analysis of this set identifies almost

[2]the point on the curve closest to the origin

50 words that would more appropriately belong in the set of semantic terms. That only three of these are regularly inflected, however, perhaps accounts for the suitability of this method in the application for which it was originally suggested.

Given that both of these techniques, and the top 100 method, simply assume some number of most frequent words as an approximation of the closed class, a more systematic approach for studying this heuristic is simply to try different partition points in the sorted vocabulary and observe the complexity estimates that result. This approach is adopted in the experiments outlined below.

5.3 Experimentation

This section outlines a series of experiments designed to measure the performance characteristics of super-adjacency models given a diverse range of predefined function word sets. The results show that super-adjacency models perform well under a variety of initial conditions, both in terms of per word entropy estimates and model size. Analysis of the final models indicates a strong correlation between performance and the number of unique instance content word bigrams, and it is hypothesised that a much smaller and more general model could be made to deliver even better complexity estimates if morphological inflection could be incorporated into the super-adjacency paradigm—a conjecture explored more fully in Chapter 6.

5.3.1 Per word entropy

A super-adjacency bigram model preserves most of the useful bigrams of a conventional model. That is, all pairs of adjacent function words and all pairs of adjacent content words normally captured as standard bigrams are also captured in a super-adjacency model. Bigrams that combine a function word with a content word, however, are reduced to one of two more general types. Function-to-content word bigrams are transformed into bigrams where the function word predicts a content category symbol, and content-to-function word bigrams are reduced to a context where the content category symbol predicts a function word. Given that the relationship between function words

and content words is primarily categorial, as argued in Chapter 4, very little contextual information is lost by such generalisation.

The advantage of a super-adjacency model is that consecutive content words which are not adjacent in a sentence can be treated as if they were. Given that close proximity content words tend to have high mutual information, the super-adjacency approach increases the availability of these low-entropy relationships. The gains that follow tend to be more than sufficient to overcome any losses from the generalisation of function-to-content word and content-to-function word bigrams. Moreover, as there are fewer possible super-adjacency bigrams than possible standard bigrams, accurate probabilities are conditioned more quickly in a super-adjacency model.

A series of experiments was devised to test the hypothesis that a super-adjacency model will outperform a conventional model in terms of its ability to deliver low estimates of language complexity. A second and more interesting objective of the experiments was to explore how changes to the initial approximation of the closed class affects the overall performance of the model. The language sample used for test input is the Brown Corpus, preprocessed in the manner described in Section 2.2. The controlled (i.e. explanatory) variable is the approximation of the set of function words, and the random (i.e. response) variable is the per word entropy given by the model.

Figure 5.3 shows a graph of the average symbol entropy for the Brown Corpus given a broad range of initial function word sets. Per word entropy, expressed in bits, is given on the y-axis, and the x-axis corresponds to the base-2 logarithm of the number of most frequent words included in the closed class approximation. There are several useful observations to be made from these experimental results.

Two forms of standard bigrams

The rightmost point on the curve plots the average symbol entropy when all 44,519 words in the Brown Corpus vocabulary are included in the function word set. The net effect is that the entire corpus is modeled using function word bigrams alone (i.e. no content category symbols are ever encountered), thus the result is equivalent to what is obtained from a conventional bigram model—approximately 6.486 bits per word.

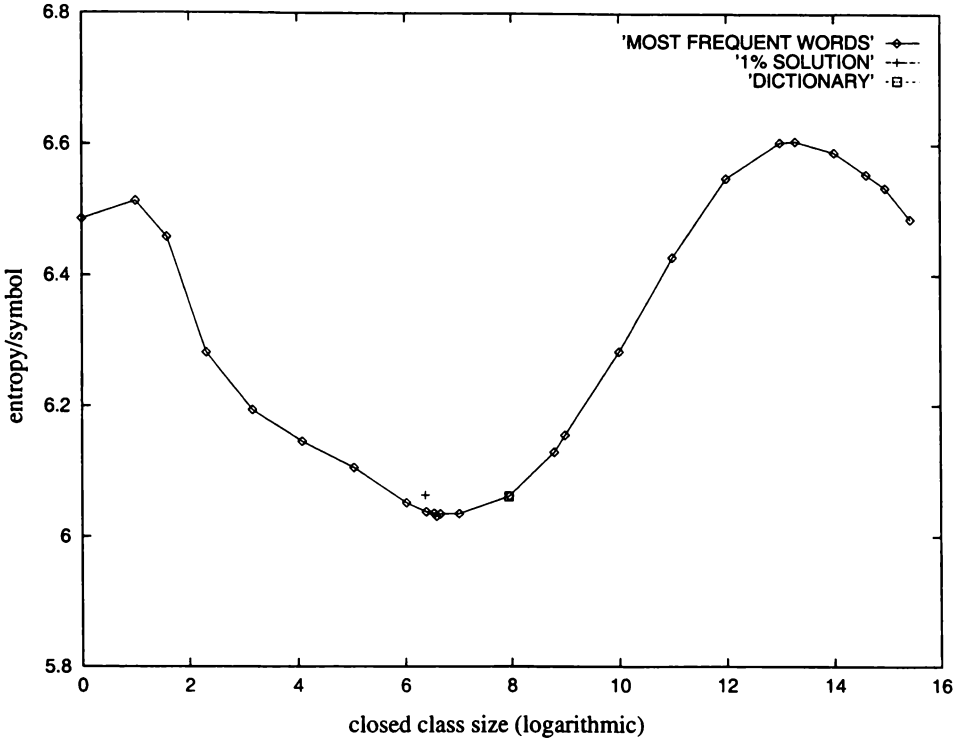


Figure 5.3: Effect of function word set on entropy/word.

Correspondingly, the leftmost point on the curve plots the average symbol entropy when no words are included in the function word set. In this instance, the entire function word stream is treated as an uninterrupted sequence of content category symbols, and every word is modeled using content word bigrams. The entropy of the category symbols is exceedingly small (effectively zero) in the functional model, and the overall result is again equivalent to what is obtained from standard bigrams.

Superior performance

The graph shows that, for function word sets comprised of between about 4 and 2000 (i.e. between 2 and 11 on the logarithmic scale) of the most frequent words, super-adjacency models of the Brown Corpus give better average complexity estimates than a conventional model. The set sizes selected for analysis where chosen to produce a graph indicative of the overall relationship between the controlled and random variables. However, the cluster

of points around the 100-word mark (i.e. about 7 on the logarithmic scale) reflects additional class sizes tested in an effort to zero-in on the optimum function word set. The best result was delivered when the 95 most frequent words were defined to be the closed class, with a per word entropy of approximately 6.03 bits—an improvement of just over 7% on the conventional bigram model.

Special cases

Also included in the graph are two additional points indicating the per word entropy from two specific closed class approximations. The 248 function words gleaned from the Oxford Dictionary resulted in an average symbol entropy of approximately 6.062 bits. As it happens, this result differs by less than 0.0003 bits from that obtained when the closed class was taken to be simply the 248 most frequently used words—but, surprisingly, this difference is *in favour* of the 248 most frequent words, indicating that a statistics-based approximation is perhaps slightly more useful when minimising language complexity is the primary goal. The second point plots the result when the 83 most frequent words from the 1% solution (Section 5.2.4) is used. This approximation produced a noticeably poorer per word entropy than the 83 most frequent words from the Brown Corpus—approximately 6.063 bits per symbol for the former as opposed to 6.038 for the latter.

Inferior models

Two other notable features of the graph are the two regions where per word entropy from a super-adjacency model is worse than for standard bigrams. For a super-adjacency model whose function word set consists of just the two most frequent symbols (the definite article "the" and the fullstop end-of-sentence marker "."), 90% of its bigrams correspond to those from a conventional model, while the rest correspond to consecutive but non-adjacent content words with presumably high mutual information. One might therefore expect this model to give at least marginally better complexity results than the conventional one, but the reality is that the average symbol entropy rises slightly to 6.513 bits.

The only explanation is that the mutual information for the super-adjacent

content terms is less than expected—specifically, it must be less than the combined mutual information each content word has with the intervening function words. It is possible to show that this is in fact what happens, but an analysis based on our own linguistic intuitions is sufficient to see *why* it happens. Consider the situation for complex nounphrases like "outcome of the experiment" and "eclipse of the moon". In a conventional model, "of the" is by far the most common bigram, but in a super-adjacency model that has just "the" and "." in the function word set, the word "of" is generalised to the content category symbol, and its predictive capacity in the syntactic model is lost. Further, the sequential relationship between "of" and the following noun is considerably weaker than the relationship between the intervening determiner and that noun, thus there is additional increase in entropy when "of" is counted among the content words. Moreover, because the content word "of" stands between two *genuine* content words with high mutual information, their semantic relationship cannot be capitalised upon because they are not super-adjacent.

When the function word set size is increased to three, "of" is incorporated into the functional sequence. As a consequence, the syntactic relationship between "of" and "the" becomes available in the functional stream, and the high mutual information of the close proximity content words is made accessible for the semantic model. Given that this form of nounphrase is so very common, the result is a significant improvement in the overall complexity estimate.

At the far right of the graph (where function word sets are larger than about 3000 words) is a much broader region where entropy estimates are worse than from a conventional model, but here the effect is much easier to understand. As the size of the function word set increases, more and more of the conventional bigrams are moved to the function word stream. As a side-effect, unbroken strings of functional terms become longer and longer so that whenever the semantic model is triggered the distance between the conditioning content word and the content word to be predicted tends to be quite large. As demonstrated in the discussion on semantic latency in Section 3.1.1, the more distant two content words are, the less likely they are to have high mutual information, and this is the observed effect in these

experiments. The penalty only disappears when the entire vocabulary is moved into the functional set and the conventional model is emulated again.

5.3.2 Model size

Super-adjacency models based on a closed class of about 100 words give better entropy estimates than do conventional bigram models, though the gains are not extreme. In the best case, based on the experiments in the preceding section, the complexity estimates from super-adjacency bigrams are just over 7% better than those obtained from standard bigrams—not much better than the 4% improvement realised by Geutner's function/content trigram model. Geutner, however, interpolated statistics of conventional word-based trigrams with those from her unconventional function/content trigrams, creating a significantly larger model than the standard approach. The super-adjacency technique, in comparison, can deliver improved performance from fewer bigrams than is required by a standard model.

Recall from Section 5.1.3 that a vocabulary comprised of F functional terms and T thematic terms leads to a comprehensive conventional model with $(F+T)^2$ bigrams. For the super-adjacency model, there are only F^2+T^2 bigrams in the worst case (although there is one additional functional term in the form of the content category symbol), thus $2 \times F \times T$ bigrams are saved. Given the 44,519 words in the Brown Corpus vocabulary, an exhaustive super-adjacency model with 100 function words has almost 9 million fewer bigrams than a complete conventional model.

Comparing the worst case model sizes of these two modeling paradigms is only of marginal interest, given that both are in $\Theta(n^2)$. What is generally more interesting is how many of the potential bigrams are actually used when modeling a large language sample. The more compact a model is, the less vulnerable it is to the problem of data sparseness, allowing accurate probabilities to be conditioned more quickly.

Compactness

Figure 5.4 plots the relationship between the size of the function word set (expressed on the x-axis and scaled to the base-2 logarithm) and the number

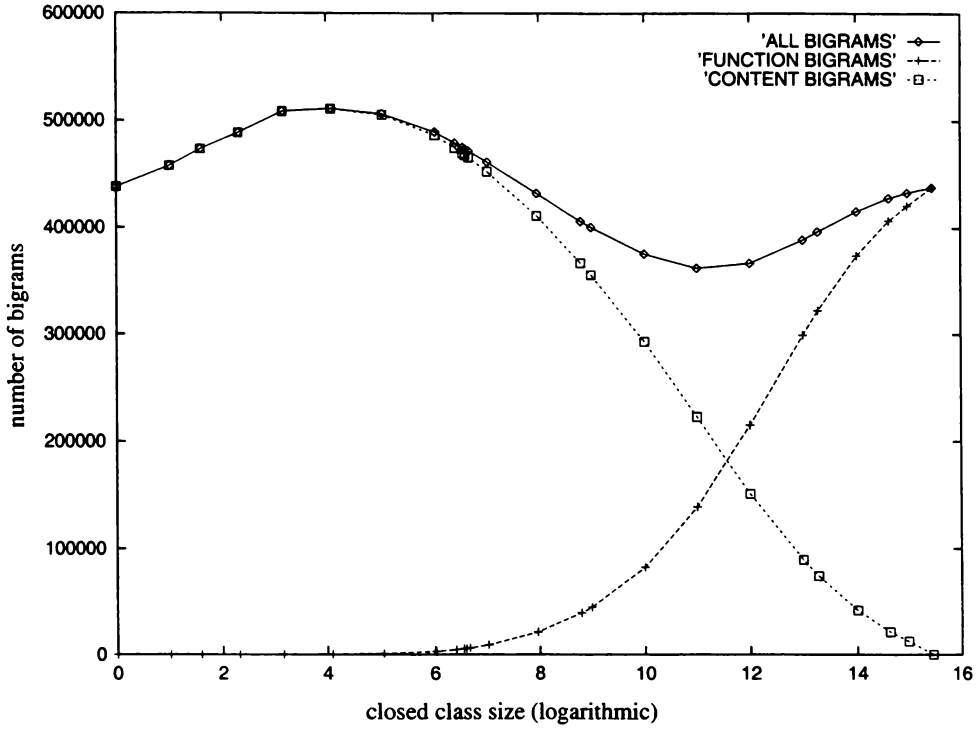


Figure 5.4: Effect of function word set on model size.

of distinct bigrams that result (expressed without scaling on the y-axis) in the corresponding super-adjacency model of the Brown Corpus. The graph shows separate curves for the number of function word bigrams and for the number of content word bigrams (the two bottom lines), and the third curve plots the sum of the other two. Once again, the end points correspond to two emulations of the conventional model—on the extreme left is the empty function word set (actually one that has just the content category symbol) and on the extreme right is the model for which the entire vocabulary is deemed the closed class.

The graph indicates that, for small function word sets, the number of bigrams in the super-adjacency model is actually greater than it is for the standard model. The worst case occurs when the 16 most frequent words constitute the closed class, at which point the model is just over 16.5% larger than a conventional one. Almost all of the excess comes in the form of content-to-content word bigrams, indicating that, contrary to initial specu-

lations, the extraction of a small number of high frequency words fails to create greater regularity in the content word stream. Viewed another way, the amount of additional mutual information made available by ignoring a small number of function words comes at the price of increased model size.

As the size of the function word set grows, the overall number of bigrams in the super-adjacency model ultimately begins to decrease, dipping below the size of the conventional model when the closed class includes around 200 of the most frequent words. Although the number of distinct function-to-function word bigrams begins to increase dramatically at this point, the rate of decrease in the size of the content word model is significantly higher, resulting in a more compact model as a whole. Interestingly, the minimum number of most frequent words needed to produce a super-adjacency model with fewer bigrams than a conventional model is 210—the knee of the curve in a graph plotting lexical rank against frequency. At this point the average per word entropy is nearly maximum at about 6.035 bits—roughly 7% lower than what the standard model gives. Thus it is confirmed that super-adjacency is able to deliver better estimates of language complexity from a smaller model.

Unique bigrams

Model compactness appears to be minimised when the function word set grows to about 2000 of the most frequent words. This is a considerably larger set of function words than is presumed for English by even the most forgiving linguistic criteria. Though it is difficult to find an explanation as to *why* this particular set size produces the smallest model, there are a number of factors that help explain *how* it happens.

The graph in Figure 5.5 plots the relationship between the size of the function word set and the number of single instance bigrams in the corresponding super-adjacency model. Similar to the previous figure, separate curves are given for the number of unique content-to-content word bigrams and for the function-to-function word bigrams, and a third curve plots the sum of these two. Included in the graph as the top line is the curve from the previous figure showing the total number of bigrams in the model. The graph implies that increased model size results directly from an increase in the number of single instance content bigrams, as the number of unique function-to-function

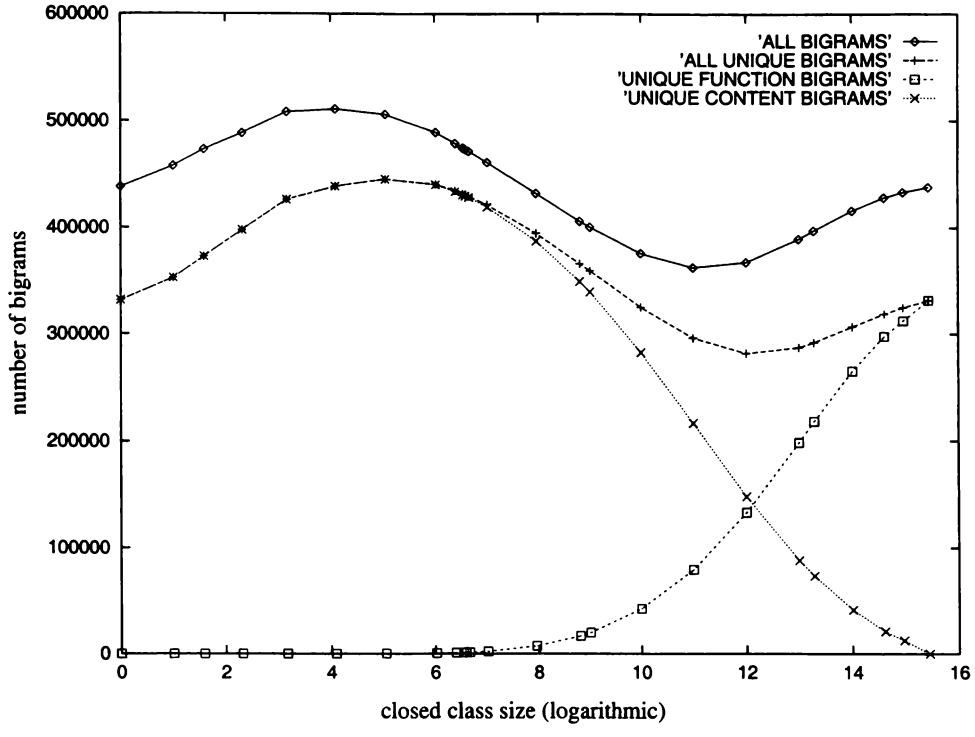


Figure 5.5: Effect of closed class size on the number of unique bigrams.

bigrams does not become notable until well after overall model size begins to decline.

The increase in unique content bigrams is also a primary cause of the decrease in average symbol entropy provided by the super-adjacency model. In the absence of smoothing, the predictive capacity of the first term in a single instance bigram is essentially perfect; thus the more single instance bigrams arising in the model, the lower the per word entropy. The effect is easier to see in the graph of Figure 5.6. In this graph, the top line shows the average symbol entropy as a percentage of the entropy given by a conventional bigram model. The bottom line plots the percentage of the model comprised of single instance bigrams. As the ratio of unique bigrams increases, the average entropy improves such that the lowest point of one curve is the highest of the other, and vice versa, indicating more or less a reciprocal relationship. Even so, for closed class sizes between 200 and 2000 words, the super-adjacency model gives superior entropy estimates from a smaller

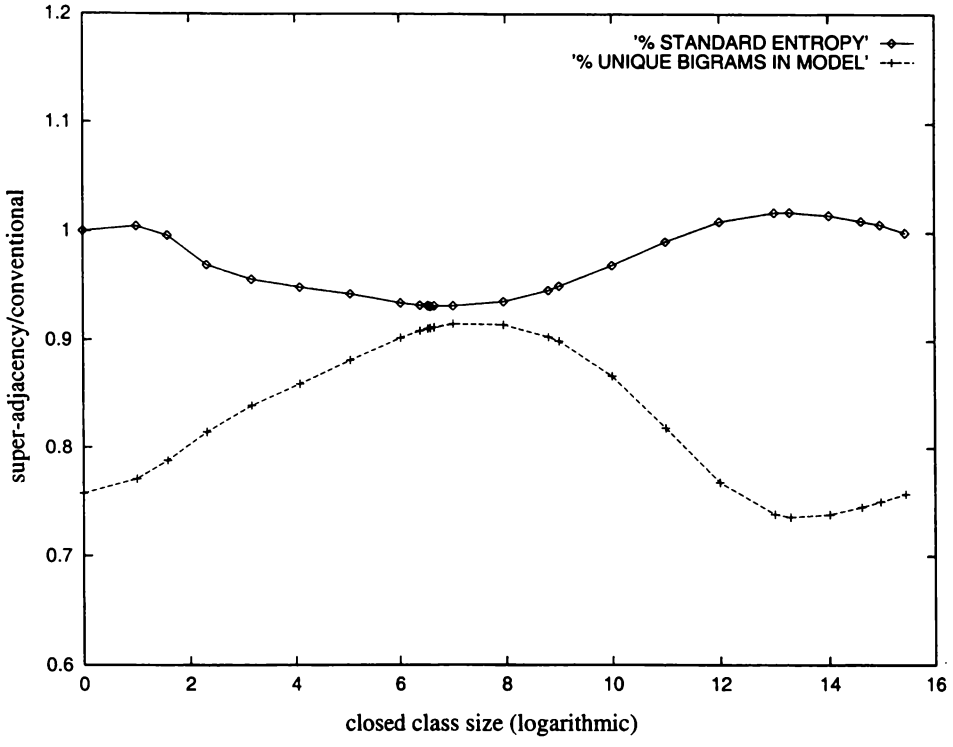


Figure 5.6: Correlation between entropy gains and the ratio of unique bigrams.

model.

5.4 Discussion

The evidence of this chapter confirms the hypothesis that the super-adjacency technique can produce better complexity estimates than the conventional approach from a more compact model. Minimum entropy is realised when 95 of the most frequent words are defined as function words. Unfortunately, the point of minimum entropy does not coincide with the point of minimum model size with respect to the size of the closed class required at their optima. Though increased availability of mutual information for super-adjacent content words is the principal factor contributing to improved entropy estimates, the fact that it appears to entail a high percentage of single instance content bigrams is somewhat unsatisfying.

The entropy results support the conclusion that the function word sequence and the content word sequence exhibit different kinds of linguistic regularity. If too many words are excluded or included in the closed class, the entropy benefits diminish rapidly, indicating that one or two hundred of the most frequent words is a suitable approximation of the closed class for making the most out of the distinct characteristics of syntactic and semantic relationships. Given such a small set of function words, it is quite feasible to increase the depth of context in the functional model, allowing more complex syntactic structure to be exploited and creating the possibility for even better per symbol entropy estimates as a consequence. However, this would do nothing to ease the problem of excess single instance content bigrams, and would in fact only add to the overall model size.

To reduce model size without entailing an increase in complexity estimates requires that the content word bigrams be generalised in such a way as to preserve their mutual information yet reduce their numbers. The following chapter outlines a means for doing just that by disentangling the core semantic root of each content word from its inflectional suffix. The result is that all subcategorisations of a content word are reduced to a base form, and the variety of bigrams involving fundamentally the same pair of core semantic terms are collapsed into a single content bigram. The inflectional suffixes are moved to the functional stream where the agreement constraints to which they are subject are better modeled as syntactic phenomena.

Chapter 6

Inflectional models

N-gram models face a considerable problem in the form of data sparseness. Given that a typical English vocabulary consists of several tens of thousands of words, the number of possible bigrams, for instance, runs into the hundreds of millions. And because lexical distributions are inherently exponential, so too is any derivative n-gram distribution, resulting in a situation where impossibly large amounts of training data are required to condition accurate statistics.

To mitigate the effects of data sparseness, a model must incorporate some form of generalisation. The super-adjacency model outlined in the preceding chapter does this to a limited extent by reducing all n-grams that involve both function and content words to one of four general cases, but the savings are rather small in comparison to what is needed to effectively combat the effects of data sparseness.

Given that function words are few in number and typically quite frequent, data sparseness is chiefly a problem attributable to content words. Thus one solution is to exploit the combinatoric characteristics of content words in a more general way. We observe that a good deal of data sparseness arises because of inflection morphology—where a general low-entropy semantic relationship between two base form thematic words is diffused by the diversity of their possible inflections. For example, in the sentences "he eats bananas", "he has eaten a banana" and "he is eating the banana", the fundamental semantic relationship between "eat" and "banana" must be modeled in three separate accounts. However, if inflected words can be glossed to their corre-

sponding base form, or *lemma*, the multiplicative effect of inflection is undone and data sparseness diminishes.

This chapter explores the possible benefits of lemmatisation for n-gram models—in particular, the effect it has on complexity estimates and model size. It is observed that lemmatisation of regularly inflected words brings about a significant decrease in the number of distinct content words and, as a consequence, large numbers of infrequent bigrams are effectively collapsed into fewer more general cases. Though much of the mutual information of semantic relationships is preserved in the lemmata, some inflection agreement information is lost and a slight increase in entropy follows. A super-adjacency model can compensate for this by moving inflectional suffixes to the functional stream where their agreement dependencies can be recaptured. The result is low entropy estimates from a much smaller model. Further, as the function word class is quite small, even when inflectional suffixes are added, it becomes feasible to consider using deeper contexts in the functional model. Access to more distant grammatical dependencies is made possible without as significant a penalty in model size as would be experienced from an equivalent higher-order conventional model.

6.1 Lemmatisation

To examine the effects of lemmatisation on the behaviour of n-gram models, it is necessary to first outline the means by which lemmata are derived. One option is to simply maintain a comprehensive list of inflected words and their corresponding base forms, but an interminable problem of out-of-vocabulary words makes this unattractive. Many rule-based stemming algorithms have been developed to circumvent this problem [73, 37, 87], but unfortunately their output is not quite suitable for the purposes of a stochastic sequential model that aims to preserve good entropy estimates from substantially fewer n-grams.

This section provides an overview of the lemmatisation problem, highlighting important concerns that suggest the need for a special purpose desuffixion algorithm. Such an algorithm is defined in the subsequent section, then applied in a series of modeling experiments detailed in the remainder of this

chapter.

6.1.1 Suffixes, stems and roots

Linguistics differentiates between two general types of suffix: derivational and inflectional. Briefly, derivational suffixes modify the major category of a word. For example, when "-less" is appended to words like "hope" or "cloud" then an adjective is derived from a noun, and when "-ly" is attached to words like "correct" or "dense" then an adverb is derived from an adjective. In comparison, inflectional suffixes mark a word for such things as tense or number (and, in some languages, also case and gender) so as to preserve syntactic agreement constraints, but the major category of the word is unaffected. For example, applying either of the suffixes "-ed" or "-ing" to an English verb can change its tense, but it remains a verb just the same.

A *stem* is a word to which a suffix has been appended,[1] so that "hope" is the stem of "hopeless" and "formalize" is the stem of "formalized". For this second example, the stem itself has a stem, "formal", which also has a stem, "form". The word that remains when no further stemming is possible is called the *root morpheme*, or often just the root. So it is that through suffixion English is able to parlay a relatively small set of lexical roots into a substantially bigger vocabulary. Conversely, this very large vocabulary can be simplified through desuffixion, reducing both the number of n-grams needed to model a language and the amount of data required to garner reliable statistics.

6.1.2 Semantic class triggers

Rosenfeld [90] explored the potential of vocabulary simplification for stochastic sequence modeling—specifically, for a model based on trigger pairs (see Section 3.2.4). His approach was to combine words sharing a common stem into a single semantic class identified by a unique *base form*—so, using Rosenfeld's example, the words "bank", "banks", "banking", "banker", and "bankers" are combined into a single class associated with the base form

[1] Linguistically speaking, a stem is a word to which an *affix* is attached, which includes prefixes and infixes, but only regular suffixes are of interest in this study.

"bank". Entropy estimates are then calculated based upon the triggering relationships between base forms. The assumption is that mutual information for two words is preserved in the associations of their respective base forms, to the extent that the triggering effect of one base form on another is similar to the triggering effect between the corresponding words. For instance, occurrence of any word with the base form "bank" ought to produce some probability boost for any word with the base form "loan", including "loans", "loaned" and even "loan" itself.

As there are substantially fewer base forms than words, there are correspondingly fewer class triggers. The expectation is that entropy estimates should improve because the generalisation allows related words to inherit the statistics of their class, reducing some of the adverse effects of data sparseness. What Rosenfeld discovered, however, was that complexity estimates from his class-based model were actually worse than those obtained from a word-based one, despite having also interpolated conventional unigram statistics into his trigger model. Rosenfeld struggled to find an explanation for the poorer performance, suggesting perhaps that when rare words combine with common words to form a class they lose the predictive power of their uniqueness without significantly improving the potential of their more frequent "classmates." This is almost certainly true in light of the results from n-gram experiments detailed in Chapter 2 and Chapter 5, where single instance bigrams are shown to play a big part in delivering low entropy estimates.

There is, however, another potential drawback for context models built from semantic base forms. Consider the following two sentences:

this bank loans its money.
these banks loan their money.

When the content words are replaced by their base forms, the sentences are reduced to

this bank loan its money.
these bank loan their money.

Generalised in this way, the number of unique contexts in a corresponding bigram model is reduced and bigram counts are increased, but the entropy

result might not improve because important agreement information is sacrificed. The base form bigram "bank loan" (where "bank" is a noun and "loan" is a verb) fails to embody the fact that a subject noun and main verb must agree in number. Similarly, the bigrams that model relationships between each of these base forms and their adjacent function words (e.g. "these bank") gives no hint as to the fact that they too must have surface forms that agree. As much as base form reduction might help diminish the effect of data sparseness, and regardless of the optimistic assumption that mutual information of semantic relationships might be preserved or even enhanced, the generalisation of n-grams in this way appears likely to entail a necessary trade-off between good entropy estimates and model compactness. To get the best of both, some means must be found to preserve important grammatical information embedded in the actual words.

6.1.3 Inflectional suffixes only

The syntactic detail lost by semantic base form reduction is predominantly agreement information formerly expressed in the suffixes of the words. One way to preserve this information without losing the potential of vocabulary simplification is to retain the suffixes in the reduced training sample as individual tokens alongside their base forms. The combined lexicon of base forms and suffixes is still much smaller than that of a complete vocabulary, but the opportunity for exploiting agreement dependencies is retained.

Isolation and retention of *all* suffixes is unnecessary—even unhelpful—and there are a number of reasons why only inflectional suffixes should be kept. Linguists maintain that inflectional suffixes are part of the functional categories of a language such that they must be learned as part of systemic grammar in order for wellformed sentences to be produced or understood. Derivational suffixes, on the other hand, are morphological components used to construct new words and may be acquired separately. And while derivational suffixes serve primarily to indicate the syntactic category of a word in isolation, inflectional suffixes participate in agreement relations with other functional terms, creating dependencies with potential for exploitation by a stochastic sequential model.

There are also good reasons to leave derivational suffixes as part of seman-

tic base forms. Derivational suffixes often have a fundamental impact on the meaning of a stem such that their detachment could interfere with modeling semantic relationships. For example, "banks" and "bankers" are certainly both associated with "loans", but bankers do very many things that banks do not, like have lunch, fall asleep at their desk, and commit suicide. Losing the distinction between various derivations might therefore undermine the ability for a word like "bank" to predict "closed" or for "banker" to predict "dead".

There is empirical evidence supporting the idea that derivational suffixes should be left alone. Xu and Croft [113] observed that semantic clustering of words based on expected mutual information measures usually results in partitions where derivations are in separate classes and inflectional forms are grouped together. As an example, they present the following set of clusters as sample output from their experiments with the Wall Street Journal.

{absorbable, absorbables}
{absorbencies, absorbency, absorbent}
{absorber, absorbers}

In this instance, the various derivations of "absorb" (apart from "absorbent") have been isolated in different classes but related inflectional forms are combined. Xu and Croft subsequently show that a base form trigram model using their semantic classes performs better in retrieval tasks than one using classes created from the conflations of an automatic stemmer.

6.1.4 Stemming objectives

To retain the utility of inflectional suffixes, some means must be found for extracting them. Using an automatic stemmer seems a practical option, but care must be taken in choosing the right algorithm. An overly aggressive stemmer that pares words down to their root can lead to excessive conflation of terms undermining the goal of accentuating perspicuous semantic relationships between base forms. For example, the widely used Porter-Lovins stemmer [87] conflates both "expectant" and "expectorate" to "expect", and all three of "precedence", "precious" and "precision" to the unhelpful (and incorrect) root "prec". Another caution is that the suffix must not be destroyed by the desuffixion process. One could try to infer the suffix afterward

by taking the difference between the stem and the original word, but this can lead to difficulties, as when "quality" is returned as the stem of "qualities". Moreover, to be most useful, suffixes themselves must be, in a way, lemmatised. Most automatic stemmers, for example, give "fox" as the root of "foxes" and "dog" as the root of "dogs". If suffixes are derived as the difference between a word and its stem, two separate plural markers result for these two words—specifically "-es" and "-s". Given that all plural markers participate in agreement relations in more or less the same way, this can lead to situations where fundamentally identical syntactic relations must be modeled with more than one n-gram.

The ideal stemmer is thus one that only processes inflected words, and whose output is both stem and suffix in some kind of canonical form. Unfortunately, stemmers of this type are not readily available, but it turns out that an appropriate algorithm can be constructed in a fairly straightforward manner. The problem is further simplified by restricting lemmatisation to regularly inflected words only, as irregular inflection does not submit to systematic treatment in any obvious way. Even people, it appears, must learn to deal with irregular forms as special cases [13, 64, 86].

6.2 A practical stemmer

This section describes a simple algorithm for reducing regularly inflected words to their lemma and suffix. The fundamental approach follows that of other rule-base stemmers, but differs in two important ways. First, it only stems a word if the result is itself a word. This avoids some of the errors that arise from overly aggressive stemming; but, more importantly, restricts stemming to situations where the simplified word will reduce the problem of data sparseness in a stochastic sequence model. Second, it produces as output both the stem and the suffix, with the suffix itself given in a general form so that its role in agreement dependencies is modeled as uniformly as possible.

6.2.1 Desuffixion

Complex English words are constructed by applying some number of derivational and inflectional suffixes to a root morpheme. But, at most, only one inflectional suffix can be added and it is always last. This property is very useful as it renders lemmatisation largely a problem of just separating an inflectional suffix from its stem. Unfortunately, simply detaching inflectional suffixes does not always produce lemmata. For example, when the inflectional suffix of "abating" is detached, what remains is "abat" instead of the correct stem "abate", and if "spanning" is lemmatised in the same way the incorrect root "spann" is left. One might elect simply to ignore problems such as these, as many stemmers do, in the hope that close enough is good enough. Alternatively, one might adopt the approach of more sophisticated stemming algorithms and apply reconstruction procedures or additional pruning to try and correct desuffixion errors. As it happens, only a few correction procedures are necessary to transform simple suffix detachment into a sufficiently robust algorithm.

6.2.2 The target suffixes

The lemmatisation algorithm detailed below addresses only five forms of regularly inflected words: nouns marked for number with a regular pluralising suffix; verbs marked for tense, number and third person singular case agreement; and nouns marked for possessive case. These restrictions greatly reduce the number of possible word endings that will trigger the lemmatisation process.

English nouns are pluralised in a great many ways. The most common is for the regular suffix "-s" to be appended to a stem, as with "dogs" and "cats". Another quite common plural marker is "-es", as with "foxes" and "kisses", which is rather more just a conventional orthographic form of "-s". There are of course a great many other less common pluralising suffixes, such as the very rare "-en" of "oxen" and the senescent transformations observed in words like "hypotheses" and "crises" or "curricula" and "bacteria". But, of all the plural marking suffixes that might be used in English, only the first two occur with any frequency as to be of practical benefit for a stochastic

sequence model of inflection.

Verbal inflection for English is much less varied and easily summarised. The suffixes of interest are simply the "-s" third person singular agreement marker used in expressions like "a dog chases" and "my uncle works", and the two regular participial endings "-ed" and "-ing" applied to words like "work" to form "worked" and "working". The fact that these endings are also used for gerunds and participial adjectives, as in "living is dangerous" and "the living planet", is inconsequential as lemmatisation is still desirable for situations such as these.

One possibly contentious inclusion in the set of targeted inflectional suffixes is the possessive marking "-'s". The reasons for including this suffix relate to the ultimate goal of isolating inflectional suffixes so that they may be more properly modeled as functional terms. The fact that many grammatical theories (e.g. DP-Theory [44]) regard possessive markers as functional terms makes it worthwhile to consider them, but of greater practical importance is the fact that they improve the performance of the stochastic sequence model used in the experiments detailed lated in this chapter. Note that it is common for nouns whose singular forms end in "s" to be marked for possessive case with just an apostrophe, and this is also incorporated into the stemmer.

Thus a set of four suffixes is defined to be of interest for this study:

$$\{ \textit{-s}, \textit{-ed}, \textit{-ing}, \textit{-'s}. \}$$

6.2.3 The stemming algorithm

The lemmatisation algorithm is very simple and is based on the same principles as other rule-based stemmers. The idea is to remove anything that looks like an inflectional suffix and then, if needed, apply some corrective procedures to the stem to turn it into a valid lemma. Unlike other rule-base stemmers, however, this algorithm only removes suffixes used to indicate regular inflection, and it produces as output both the lemma and the suffix. Moreover, rather than relying on the rules alone to make correct judgments about whether desuffixion should take place, the algorithm requires confirmation that the stem is actually a word, otherwise the suffix is left intact. The assumption is that a word is only inflected if it has been observed in an

uninflected form. This is a strong assumption (and linguistically incorrect) but it does help avoid some common desuffixion errors made by other rule-based stemmers, like mistaking "everything", "anything" and "something" as present participles. More practically, however, if a word only appears in the training text in its inflected form, then lemmatisation will not reduce the vocabulary and therefore will not help to diminish the problem of data sparseness.

Exactly what corrective procedures to apply depends in part on the kind of suffix being removed, but there are effectively only three special cases. For words like "abating" and "ceasing" an "e" must be appended after desuffixion; for words like "difficulties" and "puppies" a "y" must be added; and for words like "swimming" and "planning" the twinned consonant must be removed. The apostrophe also presents some minor difficulties, but the requisite "s" on one side or other of the apostrophe greatly simplifies things. The rules for dealing with each of the four suffixes are summarised in Table 6.1.

One added difficulty for the stemmer arises because the function word "is" is often represented in a contracted form that is identical to the possessive case marker, as in "let's go" and "Mary's gone". In such situations it is thought desirable to leave the contraction unchanged, but in the absence of sophisticated grammatical knowledge it is a difficult condition to test for. In most instances, the contraction has a function word for a stem, thus it is sufficient in practice simply to test whether the stem of an apparently possessive noun form is a function word and abort desuffixion if it is. To do this, of course, the stemmer needs to have a predefined set of function words to consult. Based on earlier experiments in this thesis, the top 100 most frequent words in the Brown Corpus was deemed a sufficiently useful approximation for this purpose.

6.2.4 Stemming results

The stemming algorithm outlined above is quite simple and, while it obviates many problems entailed from using existing stemmers, it nevertheless still produces errors. To determine precisely how robust it is as a lemmatisation procedure would require a comprehensive set of precision and recall tests, and a comparison of results with those obtained from other algorithms.

- word ends with -'
  - set stem to word less ', set suffix to *'s*
  - if stem ends with *s* and is a word then output stem and suffix, otherwise output word
- word ends with *-ed*
  - set stem to word less *-d*, set suffix to *-ed*
  - if stem is a word then output stem and suffix
  - set stem to stem less *-e*
  - if stem is a word then output stem and suffix, otherwise output word
- word ends with *-ing*
  - set stem to word less *-ing*, set suffix to *-ing*
  - if stem is a word then output stem and suffix
  - set stem to stem plus *-e*
  - if stem is a word then output stem and suffix
  - set stem to stem less *-e*
  - if stem ends with a twinned letter then set stem to stem less last letter
  - if stem is a word then output stem and suffix, otherwise output word
- word ends with *-s*
  - set stem to word less *-s*, set suffix to *-s*
  - if stem is a word then output stem and suffix
  - if stem ends with ' then
    - set stem to stem less ' and set suffix to *-'s*
    - if stem is a function word then output word
    - if stem is a word then output stem and suffix
  - if stem ends with *-e* set stem to stem less *-e*
  - if stem is a word then output stem and suffix
  - if stem ends with *-i* set stem to stem less *-i* then set stem to stem plus *-y*
  - if stem is a word then output stem and suffix, otherwise output word

Table 6.1: Lemmatisation rules.

But this stemmer is not proposed as a general purpose algorithm, rather it is designed simply to give suitable output for testing the hypothesis that lemmatisation leads to an improved super-adjacency model. The experimental results outlined below indicate that it is indeed sufficient for this purpose.

When the algorithm is applied to the complete Brown Corpus, the vocabulary of the training text is reduced from 44,519 down to 31,857, and the

number of tokens increases from 1,065,795 words to 1,200,518.

6.3 Inflection experiments

Lemmatisation of regularly inflected words in a very large corpus of English leads to reduction in the size of its apparent vocabulary and, as a result, fewer n-grams are required to model the corpus. This means that counts for individual n-grams must increase and accurate probability estimates can be conditioned more quickly, mitigating some of the effects of data sparseness. For a conventional bigram model, however, entropy estimates from lemmatised input are likely to become worse because important agreement information embodied in inflectional suffixes is lost. The goal is thus to try and retain this information in such a way that the gains in model size are preserved.

Leaving suffixes in the stream of lemmata is a possible solution, but this introduces extra tokens between semantic terms and consequently interferes with the adjacency requirement needed to capitalise on mutual information for pairs of content terms. Because agreement relationships chiefly pertain just to inflectional suffixes and function words, the super-adjacency technique offers a means to recapture agreement dependencies by moving inflectional suffixes to the closed class and modeling them as part of the functional stream.

This section details experiments using the super-adjacency technique with lemmatised input, and the results are compared against those obtained without stemming. Of specific interest is how lemmatisation affects model size and entropy estimates. The results indicate that important syntactic and semantic relationships are indeed preserved by super-adjacency when regularly inflected content words are lemmatised and inflectional suffixes are treated as free-standing function words. The net result is better entropy estimates from a smaller model than is possible from either conventional word-based n-grams or a standard super-adjacency approach.

In addition, it is shown that the super-adjacency technique makes it feasible to use deeper contexts for functional terms and thereby gain access to more long distant structural relationships. Estimates of language complex-

ity are improved, but without the same penalties associated with using a higher-order conventional n-gram model.

6.3.1 Input

The availability of both lemmata and regular inflectional suffixes suggests the possibility for modeling semantic relationships and syntactic dependencies independently in a smaller super adjacency model. The idea is to move inflectional suffixes into the closed class and model them just like any other function word, then observe what effects this has on model size and entropy estimates by repeating the experiments of Chapter 5 using lemmatised input.

The previous experiments sought evidence that the super-adjacency approach is advantageous. The basic approach was to make systematic changes to the function word set and observe the effect on model size and entropy results. Because the function word set was defined as a nonoverlapping subset of the vocabulary, repeating the experiments with lemmatisation raises a question about *when* lemmatisation ought to take place.

There are effectively two options. The input can be preprocessed so that regularly inflected words are transformed into pairs of stems and suffixes, then token statistics can be recalculated and function words chosen as before. But as larger function word sets inevitably succumb to the inclusion of more and more content terms, it is lemmata that start being treated like members of the closed class instead of the actual content words themselves. Because the frequency of inflectional suffixes leads them to be included in the closed class first, lemmata end up serving primarily as the conditioning context for their former suffixes.

If a word is functional, however, ontologically it is not inflected and should perhaps exercise its conditioning influence intact. As an alternative, then, it may be desirable to stem the words left in the open class only after the closed class has been defined. But of course this precludes the possibility of suffixes making it into the functional class, and agreement relations remain obscured. To avoid this, all regular inflection suffixes might simply be assumed functional *prima facie*, but this leads to a situation where an inflectional suffix can be both a bound morpheme (i.e. must be attached to a stem) in the functional stream and a free morpheme (i.e. a word in its own right) in the

semantic stream at the same time.

The objections to either approach are minimal, thus it was decided that the first should be adopted because it is simpler and more practical; its chief advantage being that lemmatisation of the training corpus need only be done once prior to all experiments.

Escape suffixes

A suffix must take on a kind of dual role in the model if its agreement relation with a preceding functional term is to be captured. Recall that the super-adjacency model effectively processes input from the function word model, escaping to the content word model from time to time whenever the content category escape symbol is encountered. For most closed class approximations, suffixes are treated as function words and content lemmata are treated as open class terms. This leads to the situation where every suffix is preceded by the content category symbol in the functional stream, precluding any possibility of it demonstrating an agreement relation with the most recent functional term in a way that can be captured with a bigram.

To get around this, the content category symbol preceding each suffix is simply ignored. As it is implicit that a content term must come immediately before a suffix, a small change is made to the processor so that it treats suffixes as additional content category symbols. That is, when a suffix is predicted, control passes to the content model to predict the stem that precedes it just as if the content category symbol had been detected. While this notionally upsets the sequential nature of the model, in that the suffix of a content word is predicted *before* its stem, this has no effect on the relevant model characteristics nor on how they should be interpreted.

6.3.2 Effect on model size

The experiments of Section 5.3 were repeated on the lemmatised Brown Corpus, so that for each experiment the function word set is defined as some fixed number of the most frequent words, beginning with none and increasing on each iteration by powers of two: thus the second iteration uses just the most frequent word, then the two most frequent are used, then the top four,

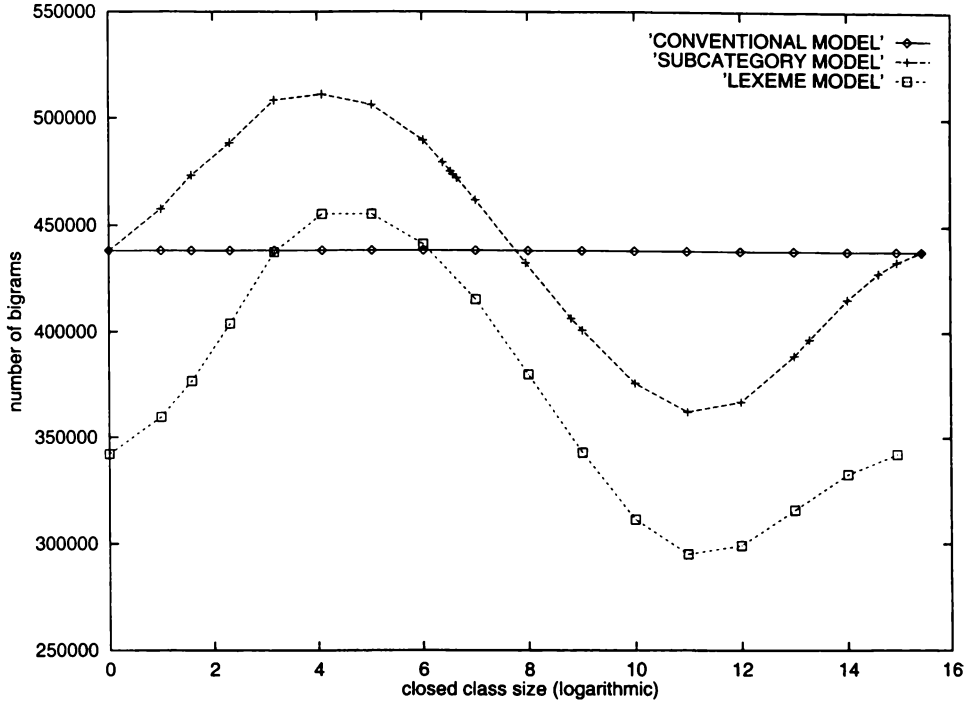


Figure 6.1: Size comparison for conventional, subcategory and lexeme bigram models.

then eight, and so on until the final experiment includes all 31,857 words of the lemmatised vocabulary.

The first characteristic of interest is how lemmatisation influences model size, and the relevant results from experimentation are summarised in the graph of Figure 6.1. As before, the x-axis is the number of most frequent words included in the closed-class (scaled to the base-2 log) and the y-axis is the number of distinct bigrams in the complete model. For comparison, corresponding model sizes obtained with the super-adjacency model when content words were still marked for subcategorisation by inflectional suffixes are included (SUBCATEGORY MODEL), along with a straight line indicating the size of a standard word-based bigram model (CONVENTIONAL MODEL). The graph shows that, for all function word set sizes, the lemmatised model (LEXEME MODEL) is much smaller than that obtained previously using word-based super-adjacency, and in almost all cases the lemmatised input gives a more compact model than the conventional approach.

The result is as expected, and it is easy to explain the behaviour of the changing model size as a reaction to changes in the size of the function word set when input is lemmatised. As with the previous super-adjacency model, the extreme left point on the curve corresponds to conventional bigrams of the lemmatised input because all words are in the content word stream and the functional stream is an uninterrupted sequence of category symbols. Compactness, relative to the other models, is as expected because the effective vocabulary has been reduced through desuffixion, leading to a corresponding decrease in the number of possible bigrams. This is enhanced from an ancillary effect brought on by the introduction of a small number of very frequent suffixes into the input stream in a highly regular way. Pairs of lemmata for content words that were formerly adjacent are now likely to be separated by the inflectional suffix of the first word. The word now acts as the conditioning context for its own suffix while the suffix itself is used as the predictor for the second word. Thus a great many low-frequency content word pairs are simplified to substantially fewer bigrams comprised of one suffix and one lemma.

When a few of the most frequent terms are moved to the closed class, model size starts to increase rapidly, as it did without lemmatisation. This is understandable because inflectional suffixes are among the most frequent terms and removing them negates some of the structural regularity introduced by lemmatisation. That is, the suffix "-s" is the second most common symbol in the input and almost on par with the most frequent function word "the". The suffix "-ed" is the fourth most common term and "-ing" is eighth, and for all closed class sizes greater than twenty-one all four of the regular inflectional suffixes are included. Just as their presence in the content word stream introduces tremendous regularity, their removal has an equally dramatic counter effect.

When the closed class gets to be sufficiently large, beginning at around 32 words, enough functional terms are removed from the semantic sequence to start reaping the same benefits as the original super-adjacency model. Specifically, close proximity content words are shunted together often enough to increase their incidence of adjacency and the number of distinct bigrams in the content model declines in response to the increased generality. Though

the functional model becomes more complex at the same time, the fact that it exhibits a great deal of sequential regularity means its size does not increase at nearly the same rate that the size of the content model diminishes. As more and more words are moved into the function word set, the model converges on an optimum size, which the graph indicates is when the function word set has around 2000 terms—roughly the same place where the standard super-adjacency model peaks. Beyond this point, adding terms to the function word set causes model size to increase again as the functional stream begins to resemble the original content stream more and more. Eventually all words are treated as closed class and the model degenerates completely to conventional bigrams, as indicated at the extreme right of the graph.

6.3.3 Entropy results

N-grams are useful because they are a simple formalism for assigning probabilities to linguistic events. Their ability to condition accurate statistics is a function of two things: the length of context used and the amount of training data available. Specifically, as the order of a model increases, the number of different n-grams increases exponentially and so does the amount of data required for adequate training. This is the fundamental problem of data sparseness.

The size of an n-gram model is only of interest insofar as it impacts on the model's ability to formulate accurate probabilities. For the conventional model these two properties tend to go up and down together, but only if availability of suitable amounts of training data is not an issue. In practice, the amount of training data is generally constant, and this places limits on how deep a context the model can sustain before the risk of finding too many novel events in subsequent testing becomes unacceptable. Thus data sparseness gives rise to the need for compromise.

The super-adjacency technique tries to circumvent this problem by using n-grams to model more abstract forms of lexical dependency in the hope that better use can be made of the available data. In Chapter 5 it was shown that this is indeed what happens—that a smaller model can be made to give better complexity estimates. Another abstraction has been proposed in this chapter, and we have shown that this leads to further reduction in the model

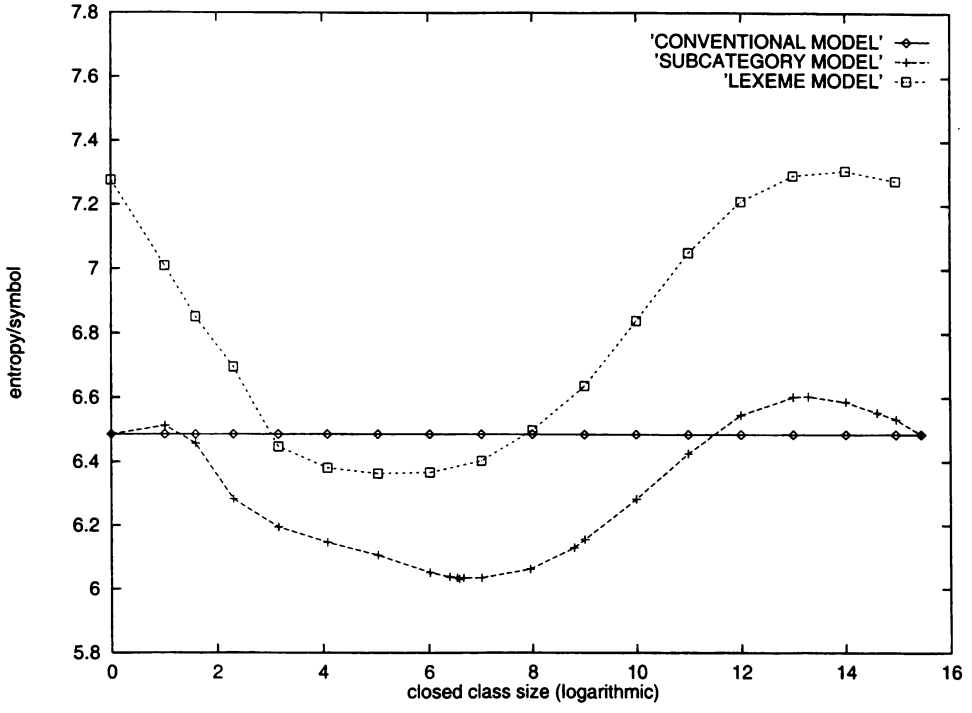


Figure 6.2: Comparison of average symbol entropy for conventional, subcategory and lexeme bigram models.

size. But if it also entails less accurate probability estimates then any notion of improvement is moot.

Figure 6.2 summarises the average per symbol entropy of the lemmatised Brown Corpus as given by super-adjacency models using various approximations of the closed class (LEXEME MODEL). Once again, for comparison purposes the entropy results given by super-adjacency without lemmatisation are included (SUBCATEGORY MODEL) along with an extended plot of the entropy given by standard word-based bigrams (CONVENTIONAL MODEL). The x-axis corresponds, as usual, to the base-2 log of the number of most frequent words defining the closed class and the y-axis is the average per symbol entropy.

As before, the extreme ends of the plot correspond to situations where the entire vocabulary is either the closed class or the functional class, rendering the super-adjacency model equivalent to a conventional one. Entropy estimates are poorer in such instances, and this is as expected because pre-

viously strong lexical relations between adjacent words become weakened by increased distance. That is, each inflected content word is separated into its lemma and suffix such that the lemma now becomes the conditioning context for predicting its own suffix. Any strong semantic relation it might once have shared with the following word is lost and it is left to the suffix alone to provide clues about the next term. Moreover, any agreement dependency between a function word and the inflectional properties of a following content word is also lost because the lemma of the next word is in the way. If an uninflected word precedes an inflected one, however, some gains might arise from the more general semantic relationship shared between lemmata, but this is unlikely to offset all other losses and the results in the graph appear to bear this out.

When a few frequent terms are taken out of the content stream, semantic relationships between lemmata are restored and entropy begins to decline in response to the sudden availability of mutual information. And in the functional model, even the top two most frequent terms, "the" and "-s", are sufficient to capture many instances of number agreement, further adding to overall entropy gains. By the time sixteen words make it to the closed class, lexical patterns that include "was" and "-ed" (e.g. "was chased") or "is" and "-ing" (e.g. "is chasing") are well captured as generalisations in the functional model.

As more and more words are transferred to the closed class, entropy eventually begins to rise again, just as it did before lemmatisation. Peak entropy occurs when the function word set has about fifty terms. Unfortunately if we consult Figure 6.1 we see that model size at this point is actually greater than that of a conventional word-based bigram model. However, there is some range of closed-class approximations where average per symbol entropy and overall model size are simultaneously superior to what is possible from the conventional model. For example, when 128 of the most frequent terms define the function word set, average entropy is 6.40 bits per symbol as compared to 6.49 for the conventional model, and the super-adjacency model has 415,013 bigrams as compared to 438,106 bigrams in the conventional model. Although the gains are quite small they are nonetheless evidence that a smaller and more effective model can be obtained by the super-adjacency technique

if regularly inflected words are stemmed and inflectional suffixes retained as functional terms. Furthermore, when the function word set includes 128 terms, the entropy results equal those obtained from a conventional bigram model but are delivered using 17% fewer bigrams.

6.3.4 Suffix obstruction

The original motivation for lemmatising the input was to try and reduce the size of the model in the hope that accurate probabilities might be conditioned more quickly. But we recognised ahead of time that complexity estimates would likely get worse unless important agreement information embedded in inflections could somehow be preserved. It was conjectured that retention of inflectional suffixes as individual tokens could present the opportunity to do this, but only if the relationships in which they are involved are specifically targeted—say, by treating suffixes as if they are function words in their own right.

The graph of Figure 6.2 shows that this approach is successful; that the initially poorer complexity estimates are quickly brought under control again when just a few of the most common words are moved into the closed class. In fact, entropy results become better than those obtained from a conventional model when as few as eight terms make it into the functional set. But the super-adjacency model never quite manages to attain the same level of performance as it did before lemmatisation; thus some predictive potential has been lost.

One likely explanation for this decrease in performance is that the presence of inflectional suffixes is interfering with more useful syntactic dependencies that existed previously between *bona fide* function words. There is some evidence to support this view. Consider, for example, the following set of English subexpressions:

distribution of funds is ...
appointment of administrators is ...
surveillance of prisoners is ...
study of insects is ...

Without lemmatisation, these expressions have a common functional structure that might be characterised as

CW of CW is ...

and this general pattern is relatively common in the Brown Corpus, arising about 500 times. In this sequence, "of" predicts the content category symbol CW, and the category symbol predicts "is". Linguistically, there is no *direct* relationship between "of" and "is" because "of" is the head of a prepositional phrase qualifying the preceding noun, and it is that noun's number that must agree with "is". With lemmatisation, however, the common functional structure becomes

CW of -s is ...

Now the word "of" predicts the inflectional suffix "-s", and the suffix predicts "is". Given that "-s" is a plural marker it might normally be expected to predict "are" instead of "is", thus lemmatisation has, in this instance, confounded an important agreement relationship.

There are many similar situations where a weaker or incorrect relationship is modeled because an inflectional suffix has been introduced into the stream (e.g. try adding "by experts" just before "is" in the previous examples), indicating that perhaps an entirely separate model of the sequential characteristics of inflectional suffixes may be required to maximise their utility without decreasing the utility of actual function words. However, the heart of the problem may have more to do with the fundamental limitations of order-1 contexts when it comes to generalising the complexities of syntactic structure.

6.3.5 Single instance bigrams

From a practical perspective, the fact that lemmatisation offers such small gains is disappointing. But there is some evidence to suggest that the model is actually a more effective generalisation of lexical dependencies.

Section 5.3.2 describes how single instance bigrams have an enormous impact on a model's overall ability to generate good complexity estimates. In general, the more unique bigrams there are, the lower the entropy; but a high percentage of single instance bigrams is also a sign that the model is over-fitting the data and is thus not a good generalisation of the underlying dependencies.

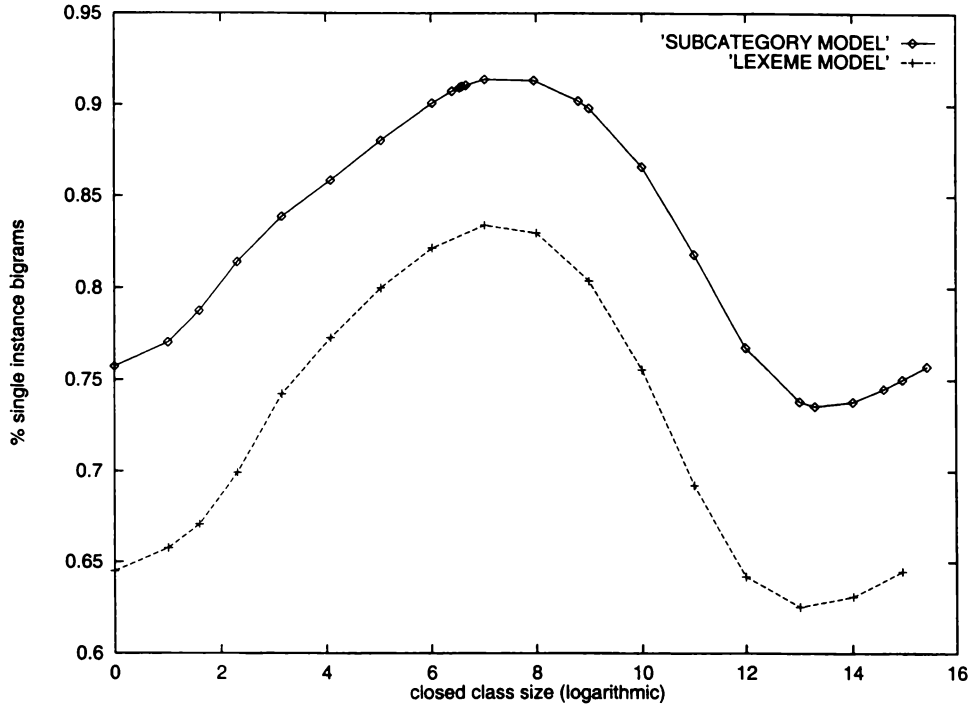


Figure 6.3: Percentage of single instance bigrams in subcategory and lexeme models.

Figure 6.1 shows that lemmatisation leads to a smaller model, and Figure 6.2 shows that better entropy estimates are achieved. This evidence alone suggests that the inflectional model is a more effective generalisation. But it is also possible that the utility of a small number of very common bigrams has improved enough to account for all of the gains, and that the model still maintains an extremely large number of special case contexts.

The graph in Figure 6.3 plots the percentage of single instance bigrams in the super-adjacency model when it is trained on lemmatised input (LEXEME MODEL) as compared to when the input is not lemmatised (SUBCATEGORY MODEL). The graph shows that lemmatisation significantly reduces the number of special case contexts needed to account for lexical patterns in the data. For all approximations of the function word set, the improvement is about 10% relative to the overall model, and about 15% relative to the number of unique cases before lemmatisation. At the point where model size and complexity estimates are simultaneously better with lemmatisation (i.e.

when there are 128 words in the closed class) there are well over 50,000 fewer single instance bigrams. These results are a clear indication that lemmatisation allows for a much more general characterisation of lexical dependencies with the super-adjacency technique.

6.3.6 Deeper contexts

As noted earlier, the primary objective of an n-gram model is to provide accurate predictions about linguistic events, and data sparseness is a significant limiting factor. Super-adjacency is proposed as a means to moderate this problem by exploiting more useful forms of lexical dependency, and the experiments above show that lemmatisation of the input further increases its ability to converge on optimum probability estimates more quickly than the conventional approach. But there are times when even rapid convergence on a theoretical optimum is not sufficient for practical language processing tasks, and more accurate probabilities are needed than can be obtained simply by increasing the amount of available training data.

The most obvious way to improve the entropy results from an n-gram model is to increase the length of context used to predict each symbol—that is, use a higher-order model. But this greatly exacerbates the problem of data sparseness. As context length increases linearly, model size increases exponentially and so too does the effective minimum data requirement. The exact rate of growth for practical models is difficult to determine in general, but the worst case for an order-n model with v vocabulary terms is v^n. For the 44,519 different words of the Brown Corpus, a bigram model has to maintain statistics for almost two thousand million parameters. An increase to an order-3 model creates the potential for 90 million million distinct trigrams. Super-adjacency, however, provides an opportunity to mitigate this problem as well.

The super-adjacency technique entails partitioning the vocabulary into two classes: a closed class with (optimally) around 100 words, and an open class with several tens of thousands of words. Because each class is modeled independently, it is possible to increase the length of context used in one while maintaining statistics for shorter contexts in the other. Given that the closed class is so small, its order can be increased to gain access to at least

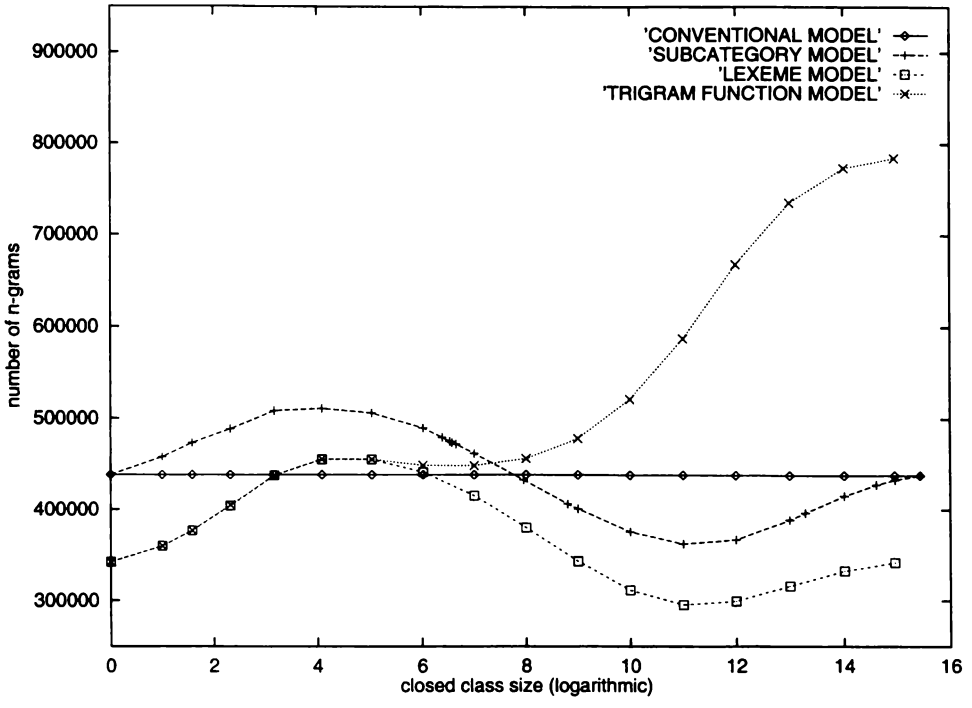


Figure 6.4: Effects of function word trigrams on model size.

some of the more distant lexical dependencies without nearly the same cost in overall model size and data requirements as would be entailed by increasing the order of the whole model.

Figure 6.4 provides a graph showing the effect on super-adjacency model size when function word trigrams are incorporated (TRIGRAM FUNCTION MODEL). At the extreme left, only the content category symbol is in the closed class and therefore only one trigram is used (which predicts with perfect accuracy), thus the model is effectively a conventional bigram model using lemmatised input. As the approximation of the closed class increases in size, the number of function word trigrams increases more quickly than it does when bigrams are used, but because the number of function words is so very small the additional cost is undetectable in the graph.

Given the scale and resolution of the graph, model size differences need to reach about 5,000 before we would expect to be able to see separation for two otherwise colinear curves. A comprehensive functional model with sixteen words and one content category symbol would have 289 bigrams or

4,913 trigrams—a difference that should perhaps be noticeable in the graph. As it happens, while the bigram model at this point has 266 bigrams, the trigram model has only 1406 and no separation is observed in the graph. When the closed class reaches 64 words plus the content category symbol, worst case scenarios for each model would yield 4225 bigrams and 274,625 trigrams, and the difference should be quite visible. Using the lemmatised Brown Corpus as input, however, the number of bigrams is almost exhaustive at 3193, while the 20,175 trigrams is less than a tenth of the total possible, and the difference in model size is only just visible in the figure.

These results suggest that the sequential behaviour of function words is highly constrained, and that there is not a lot of variation in the way function words combine together in syntactic structures. Even when there are 256 words in the closed class, plus the content category symbol, there could be as many as 17 million different combinations of three function words, but only 118,103 of these are observed in the Brown Corpus—barely half of one percent. Only when the closed class starts to include actual content words are the exponential consequences of higher order contexts realised in the super-adjacency model. The implication is that a great deal of syntactic regularity is indeed being captured when *bona fide* function words are modeled in isolation.

The fact that functional trigrams do not exhibit much more variety than functional bigrams suggests that perhaps only marginal gains in entropy will follow from the increased context. That is, if the number of trigrams is not significantly higher than the number of trigrams for a given closed class, then the bigrams may be capturing the sequential characteristics sufficiently well.

Figure 6.5 compares the entropy results from function word trigrams (TRIGRAM FUNCTION MODEL) against those obtained from the other three models. With no words in the closed class, the model with trigrams is effectively the same as the bigram model with lemmatised input and the entropy results are the same. But even when just one common word is included in the set of function words, entropy results begin to diminish more quickly with the availability of trigrams. By the time five or six of the most common terms are moved to the closed class, the model performs on par with a super-adjacency model that does not use lemmatised input. This may suggest that

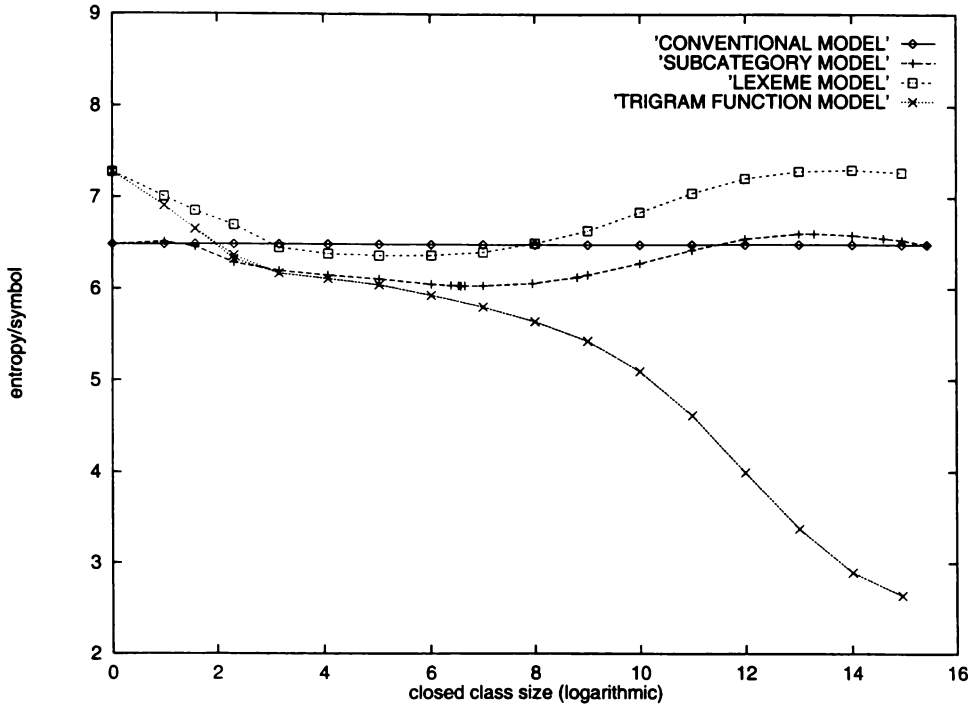


Figure 6.5: Entropy when function word trigrams are combined with content word bigrams.

the inflection agreement information that was lost through lemmatisation and only partly regained by modeling inflectional suffixes with functional bigrams is more or less completely recaptured when trigrams are available. Entropy gains continue to improve slightly through larger and larger approximations of the closed class, but significant improvement only starts to become apparent when content terms inevitably start to be included and the model begins its steady degeneration into a conventional trigram model.

6.4 Discussion

In Chapter 5, the super-adjacency model was proposed as a means for separating the problems of modeling syntactic relations and semantic relations by characterising the sequential behaviours of function words and content words in isolation. This chapter has extended the idea in such a way that syntactic and semantic associations between two content words may also be

treated independently. It has been argued that the semantic dependency between two content words has to do with an association of meaning conceptually embedded in their semantic base forms, while syntactic agreement dependency is an effect entirely attributable to their inflectional component. The solution is to lemmatise any regularly inflected content word and treat its inflectional suffix as just another syntactic term.

This chapter has shown that the separation of stem and suffix for regularly inflected content words does lead to a more effective super-adjacency model. The bulk of the mutual information for a pair of content words is preserved in their lemmata and, because lemmatisation greatly simplifies the vocabulary, the number of content bigrams observed in a training text diminishes greatly. This leads to general increases for bigram counts and, therefore, reliable probabilities are conditioned more quickly.

The probabilities are reliable, but not necessarily better for obtaining good estimates of language complexity because all syntactic agreement constraints are lost when inflectional suffixes are detached. Given that syntactic agreement is a property of the functional categories of language, the predictive benefits of inflectional suffixes can be recaptured by treating them as free-standing function words. The experimental results outlined in this chapter have shown that this approach allows for good complexity estimates from a much more compact model.

At some point, any bigram model eventually maximises its predictive capacity in the sense that additional training data is not going to improve its entropy estimates in any significant way, and this is no less true for a super-adjacency model. The fact that lexical dependencies often exist across greater distances than bigrams can accommodate ultimately leads to a situation where lengthening the predictive context is the only option for achieving further gains. However, a general increase in the order of the model is often unsuccessful because there are exponential consequences in terms of data sparseness. But because the super-adjacency technique models function words in isolation, and there are relatively few of them, the use of deeper contexts in the functional model alone *is* viable, and this chapter has shown that better entropy estimates result without a significant penalty in terms of overall model size.

Chapter 7

Conclusions

N-gram models are a fundamental component of many practical language processing systems. Their appeal is overall simplicity and an ability to support reasonably sound decisions in the face of uncertain linguistic events. Even so, conventional word-based n-grams are fundamentally limited in terms of the kinds of linguistic regularity they can access, and their exponential model size leads to an inherent problem of data sparseness, where impossibly large amounts of training data are required to condition accurate probabilities.

This thesis has shown that these problems can be mitigated by making a distinction between function words and content words and modeling the sequential behaviours of each independently. The idea is to try and isolate syntactic and semantic dependencies in such a way as to create the opportunity for a more effective characterisation of structure in language. Moreover, by reducing regularly inflected content words to lemmata and treating their inflectional suffixes as free-standing functional morphemes, mutual information for words involved in semantic relationships can be exploited with far fewer n-grams, and agreement dependencies can be modeled in a more general way. The super-adjacency model presented in this thesis embodies these ideas, and we have shown that it provides better estimates of language complexity from a much smaller model than is possible from a conventional approach.

In this chapter, we summarise important observations and achievements detailed earlier in the thesis, and consolidate them into terse statements

highlighting the contributions made to language modeling research. Avenues for possible future work are also outlined, along with some closing remarks.

7.1 Summary

Statistical models have proven very effective at a variety of specific language tasks, such as speech recognition, text compression, part-of-speech tagging and information retrieval. By viewing language as a stochastic process, a system can be conditioned to make a good guess as to how to proceed when confronted with novel input. N-gram models are the preferred formalism, primarily because of their simplicity, but also because they have potential to be made at least as effective as any other conceivable model. In practice, their performance is severely limited because the scale of model required creates an insurmountable problem of insufficient training data. In fact, even low-order models trained on very large samples frequently find themselves confronted with contexts for which they have just not seen enough evidence to make the correct decision.

One solution is to direct n-gram models away from the onerous task of collecting statistics about every conceivable combination of words and instead gear them up to focus on more abstract forms of linguistic dependency as suggested by syntactic theory. The primary hypothesis of this thesis is that separate models of the sequential characteristics of function words and content words is a viable way to do this, and that better estimates of language complexity will result from a smaller model.

Initial evidence in support of the conjecture is found in an analysis of results obtained through experimentation with conventional word-based n-grams. The general utility of a particular n-gram is measured by its average expected entropy, and from this it is observed that most gains realised by the conventional model are made from a relatively small number of frequent n-grams comprised exclusively of grammatical terms, combined with a large number of rare n-grams comprised solely of content words with strong semantic relationships. Added to these is a plethora of n-grams that include both grammatical and content words but which fail to advance the predictive goals of the model in a generally useful way.

Additional support for separate characterisation of function and content words is derived through experiments with lexical attraction models. Such models assign structure to sentences by linking words in a manner that maximises the total available mutual information without violating planarity for the parse tree. They subsume bigram models, but have the potential to outperform them because any word in a sentence can act as the conditioning context for any other. As it happens, semantic relationships tend to dominate the lexical dependency structure, and content words end up being linked to one another regardless of how far apart they are in a sentence. In comparison, mutual information for lexical relationships involving function words is generally quite minimal, and because of the planarity constraint they end up being tacked on with short links to nearby words, often to each other. Inasmuch as dependency structures seek to maximise total mutual information, the net effect is a tendency for content words to link to each other and for function words to link to each other, and for connections between these two broad classes to be largely subordinate in establishing the final structure.

A potential hazard for separate modeling of function and content words is total loss of information about dependencies between the two classes—for example, the relationship between determiners and nouns in constituent noun-phrases. Because dependencies between function and content words appear to operate at the level of lexical categories—in that determiners can predict nouns but not specific words—some experiments were devised with a class-based n-gram model to explore the characteristics of categorial relationships. The results suggest that highly discriminate function word categories, to the extent of one word per category, and very general content word categories, to the extent of one category for all content words, is largely sufficient to maintain adequate predictions about how function words and content words interact. Moreover, it is observed that even a very crude approximation of the function word class based solely on lexical frequency is satisfactory for delivering good entropy estimates from category-based n-grams, and this obviates the need for an *a priori* classification scheme.

The super-adjacency model is proposed as a mechanism for giving n-grams better access to syntactic and semantic dependencies. The model views language as two interlaced streams—one comprised only of function

words and the other of content words—and each is modeled independently. Interaction between the two streams is coordinated by inserting a unique content category symbol in the stream of function words at each point where a content word must be predicted. The probability of a function word is made conditional on the most recent function word, and whenever a category symbol is encountered the next content word is predicted based on the context of the most recent content word. The result is a 10% improvement in the average per symbol entropy estimates over what can be obtained from a conventional model.

The best entropy estimates from a super-adjacency bigram model occur when the function word class includes around a hundred of the most frequent words, and the model size at this point is slightly smaller than the conventional model. Optimum size, on the other hand, occurs when the function word class is comprised of about a thousand most frequent words, and average entropy at this point is just slightly below what is given by standard bigrams. For all class sizes in between, entropy estimates and model sizes are simultaneously improved. This confirms the principal claim of the thesis, that separate modeling of function words and content words can give better entropy estimates from a smaller model.

The fact that optimum performance does not coincide with optimum size gives rise to the idea that the basic super-adjacency technique might be improved. It is observed that much of the entropy gains come from a large number of single instance bigrams, and that these arise in part because of the multiplicative effects of inflection morphology. On the assumption that the relationship between two content words is primarily semantic, the bulk of their mutual information can be preserved in fewer bigrams by lemmatising inflected words. To preserve any loss of useful agreement information, the inflectional suffixes are retained as part of the functional stream—an idea that is consistent with the linguistic view that inflectional morphemes are part of the functional categories of language. Because agreement relations often exist between inflectional suffixes and function words, the new model is on the whole better able to exploit syntactic dependencies. The result is an even more compact model that delivers better estimates of language complexity.

Finally, because the function word set is quite small even with the added inflectional suffixes, the use of deeper contexts for the function word stream becomes feasible. This allows some of the benefits of more distant agreement relationships to be exploited by the super-adjacency technique without as significant a penalty in model size as would be entailed by an equivalent higher-order conventional model, and this is substantiated by experimental results.

7.2 Contributions

This thesis makes a number of contributions to language modeling research. In approximate order of importance they are:

- evidence that dependencies between function words and between content words are generally more useful for stochastic sequence modeling than relations between function and content words; specifically
  - conventional n-gram models make most of their gains from n-grams comprised solely of function words or solely of content words;
  - dependency structures based on lexical attraction tend to prefer links between content words first, between function words second, and lastly between content words and function words;
- an effective heuristic for finding lexical attraction dependency structures that make good use of the total available mutual information but entail no cost for specifying the structure itself;
- the finding that increased specialisation of function word categories increases the availability of class-based dependencies, while extreme generalisation of content word categories has few negative effects;
- a framework for modeling the sequential characteristics of function words and content words in isolation—namely, the super-adjacency technique;

- the finding that an approximation of the closed class based solely on high lexical frequency is as effective for stochastic sequence modeling as a more considered class definition; specifically
  - compression from unbounded category contexts gives almost identical results when words are labeled with a crudely derived frequency-based tagging scheme and when they are tagged with the more discriminate AMALGAM tagger;
  - entropy estimates from a super-adjacency model which defines the function word set solely based on lexical frequency are not surpassed by a model whose function word set has been derived from a dictionary;
- separate bigram models of the sequential characteristics of function words and content words allows better entropy estimates to be obtained from a smaller model than is possible from the conventional approach;
- a simple algorithm for lemmatisation of regularly inflected words;
- the finding that inclusion of inflectional suffixes in a separate model of functional terms, combined with a model of semantic lemmata, leads to better entropy estimates from substantially fewer n-grams than is possible from a strictly word-based approach;
- the finding that deeper function word contexts improve entropy estimates of language but do not entail a significant cost in terms of increased model size;

7.3 Future work

Stochastic language models are applied with considerable success to a wide variety of practical language processing tasks, and the quest for ever better probability estimates of specific linguistic features continues as one of the more active areas of computer science research. The super-adjacency model outlined in this thesis demonstrates that even the basic n-gram approach has untapped potential for providing better performance, but a number of issues have been raised suggesting further research.

Of primary importance is the need to determine just how robust the super-adjacency technique is. Models based upon conditional probabilities are susceptible to making invalid assumptions about what is known or needs to be known when estimating a probability, and one way to test whether or not such assumptions are made by a particular formalism is to apply it to a practical language processing task where overlooked information cannot go unnoticed. Text compression, for example, reduces plain text to a size proportionate to the probability assigned to it by the compressor's underlying model. If the compressor's output can be restored to its original uncompressed form without loss of detail and without any additional information then soundness of the approach is confirmed.

Some effort has already been given towards the construction of a text compression scheme based on super-adjacency. An adaptive algorithm has been trialed that uses a super-adjacency model without lemmatised input. Preliminary experiments have delivered compression ratios commensurate with what is expected based on the experiments detailed in Chapter 5. And a complementary decompression scheme has been developed that restores the compressed language sample to its original form, providing evidence that the results outlined in this thesis are accurate. Moreover, the compressor achieves its results adaptively, using statistics conditioned "on the fly", and this indicates that the use of posterior probabilities in the calculations of this study has not overly distorted the results. The fact that the compressor gives the expected results for a variety of sample inputs further suggests that the approach is robust. It remains to be seen now whether lemmatisation can be incorporated with similar success.

Because the super-adjacency model maintains different models for grammatical terms and semantic terms, it may prove a useful tool for certain information retrieval and text mining tasks. For example, a super-adjacency model can be trained on a collection of documents pertaining to a particular topic, then the semantic model alone could be used to calculate entropy estimates for the content word stream of unknown documents. Those addressing similar topics would likely yield low cross entropy scores while unrelated documents would give high scores. Conventional n-grams can of course do the same thing, but the fact that they make most of their gains from function

word n-grams leads to greater ambiguity in their results. Alternatively, if one wanted to be able to distinguish whether a source was a technical work or a novel, one might train a super-adjacency model on a collection of documents of one kind or the other and then just use the grammatical model to calculate cross-entropy scores for the functional sequence in an unknown document. Given that technical writing typically employs quite a different grammatical style than fiction, a high entropy score may be a reliable way to determine a match.

Chapter 4 provides evidence that the distinction between different content word categories is not terribly important for a class-based context model, and throughout this thesis there have been suggestions that further subcategorisation of content words is implicit in other ways. Still, the fact that there are very real dependencies between, say, determiners and nouns, and between modal auxiliaries and verbs, suggests that the use of a single content word category symbol may be an excessive simplification. A key factor leading to this simplification is the absence of a clear cut method for subclassing content words. Lexical frequency was shown to be suitably effective for identifying function words, but more sophisticated techniques may be needed to differentiate between specific kinds of content words.

In the lemmatisation model, inflectional suffixes were also used as content category symbols. Given that verbs inflect in manners that nouns do not, suffixion may offer some clue for distinguishing between these two categories, and this is something that could be explored. Even incorporating the inflectional suffix into the context for predicting its associated stem seems a logical option to examine. In fact, regular derivational suffixes typically signal the grammatical category of a word, and consideration of these may also be beneficial for developing a more discriminating classification scheme within the model.

The content category symbol is troublesome in other ways. The fact that some number of content category symbols frequently stand between function words involved in a dependency relationship makes it desirable to seek a way to reach across this distance. Longer function word contexts have been shown to help, but this approach does not generalise over syntactic patterns of variable length. We suggested earlier that separate statistics might be

maintained so that a function word can be predicted using the context of the most recent function word regardless of any intervening content category symbols, and this is a certainly a modification that merits further study.

One of the most interesting observations from Chapter 6 was that the number of function word trigrams was not significantly larger than the number of function word bigrams for closed class sizes smaller than about 128 words (see Figure 6.4). It would be interesting to see if this holds for higher-order contexts. If so, it may suggest that grammatical patterns could actually be rote learned.

It was claimed at the outset that this thesis is not presenting a theory of grammar or learning, but is instead addressing the problem of developing a sound performance-based model of language. However, it was also stated that consideration to theories about genuine linguistic phenomena is a design motivation. Whether the super-adjacency technique upholds this objective is difficult to determine in light of the fact that all of the experiments have been directed specifically at English, and it may be of theoretical interest to try to adapt the model to work for other languages. In any event, the goals of language modeling are not restricted to practical problems, nor is interest in the topic limited to computer scientists. The basis, framework, and results from any particular model have the potential to provide insights into more general questions about sequence modeling, automatic induction, grammatical theory, learnability and even cognition.

References

[1] Steven Abney. Functional elements and licensing. Presented to GLOW, Gerona, Spain, April 1986.

[2] Steven Abney. *The Noun Phrase in its Sentential Aspect.* PhD thesis, MIT, Cambridge, Mass., 1987.

[3] N. Abramson. *Information Theory and Coding.* McGraw-Hill, New York, 1963.

[4] Diana B. Archangeli and D. Terence Langendoen. *Optimality theory: an overview.* Blackwell Publishers, Malden, Mass., 1997.

[5] J. Bach and I. H. Witten. Lexical attraction for text compression. In J.A. Storer and M. Cohn, editors, *Proceedings of DCC '99.* IEEE Computer Society Press, 1999.

[6] L. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2), March 1983.

[7] Doug Beeferman, Adam Berger, and John Lafferty. A model of lexical attraction and repulsion. In *Proceedings of the ACL-EACL '97 Joint Conference*, Madrid, Spain, 1997.

[8] T.C. Bell, J.G. Cleary, and I.H. Witten. *Text Compression.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[9] T.C. Bell, I.H. Witten, and J.G. Cleary. Modeling for text compression. *Computing Surveys*, 21(4):557–591, December 1989.

[10] J. L. Bentley, D. D. Sleator, R. E. Tarjan, and V. K. Wei. A locally adaptive data compression scheme. *Communications of the Association for Computing Machinery*, 29(4):320–330, April 1986.

[11] D. V. M. Bishop and S. J. Bishop. Twin language: A risk factor for language impairment? *Journal of Speech, Language, and Hearing Research*, 41(1):150–160, February 1998.

[12] Robert D. Borsley. *Syntactic Theory: A Unified Approach.* Edward Arnold, Sevenoaks, Kent, 1991.

[13] M. Bowerman. How do children avoid constructing an overly general grammar in the absence of feedback about what is not a sentence? *Papers and Reports on Child Language Development* 22, Stanford University, 1983.

[14] Joan Bresnan. An approach to universal grammar and the mental representation of language. *Cognition*, 10:39–52, 1981.

[15] Joan Bresnan, editor. *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Mass., 1982.

[16] Chris Brew. Stochastic HPSG. In *Proceedings of EACL-95*, 1995.

[17] E. Brill and M. Marcus. Automatically acquiring phrase structure using distributional analysis. In *Darpa Workshop on Speech and Natural Language*, Harriman, N. Y., 1992.

[18] P. Brown, P. V. de Souza, R. Mercer, V. J. Della Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 8(4), 1992.

[19] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40, 1992.

[20] D. Caplan. *Neurolinguistics and Linguistic Aphasiology.* Cambridge University Press, Cambridge, 1987.

[21] R. Carnap. *Logical Foundations of Probability.* University of Chicago Press, 1950.

[22] S. F. Chen. *Building probabilistic models for Natural Language.* PhD thesis, Harvard University, Cambridge, Massachusetts, Cambridge, Mass., 1996.

[23] N. Chomsky. *Syntactic Structures.* Mouton, The Hague, 1957.

[24] N. Chomsky. On certain formal properties of grammars. *Information Control*, 2:137–167, 1959.

[25] N. Chomsky and G. Miller. Formal analysis of natural languages. In Luce, Bush, and Galatier, editors, *Handbook of Mathematical Psychology II*, pages 269–593. Wiley, New York, 1963.

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

[27] Noam Chomsky. *Rules and Representations.* Oxford: Blackwell, 1980.

[28] Noam Chomsky. *Lectures on Government and Binding.* Foris Publications, Dordrecht, 1981.

[29] Noam Chomsky. Some notes on economy of derivation and representation. In Itziar Laka and Anoop Mahajan, editors, *MIT Working Papers in Linguistics 10: Functional Heads and Clause Structure.* Department of Linguistics and Philosophy, MIT, 1989.

[30] Noam Chomsky *The minimalist program.* MIT Press, Cambridge, Mass., 1995.

[31] J. G. Cleary and I. H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):396–402, 1984.

[32] Alain Colmerauer. Metamorphosis grammars. In L. Bolc, editor, *Natural Language Communication with Computers*, pages 133–189. Springer Verlag, Berlin, 1978.

[33] D. Conklin and I. H. Witen. Complexity-based induction. *Machine Learning*, 16(3), 1994.

[34] G. G. Coulton. *St. Francis to Dante.* David Nutt, London, 1906.

[35] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language (ACL)*, Trento, Italy, 1993.

[36] Flores d'Arcais. Lexical knowledge and word recognition: Children's reading of function words. *Visible Language*, XVIII(4):353–371, Autumn 1984.

[37] J. L. Dawson. Suffix removal and word conflation. *ALLC Bulletin*, pages 33–46, 1974.

[38] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society on Information Science*, 41(6):391–407, 1990.

[39] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis.* Wiley, New York, 1973.

[40] T. Díez E. Luque, A. Ripoll. Vertical migration: An experimental study of the candidate selection problem. *IEE Proceedings, Part E, Computer & Digital Technology*, 134(4):177–188, 1987.

[41] Andreas Eisele. Towards probabilistic extensions of constraint-based grammars. Deliverable r1.2.b, DYANA-2, September 1994.

[42] David Elworthy. Does Baum-Welch re-estimation help taggers? In *Proceedings of ANLP-94*, Stuttgart, 1994.

[43] Steven Finch and Nick Chater. Bootstrapping syntactic categories using statistical methods. In D. Daelemans, W. & Powers, editor, *Background and Experiments in Machine Learning of Natural Language*, pages 229–236, Tilburg, NL., 1992. ITK.

[44] Naoki Fukui and Peggy Speas. Specifiers and projection. *MIT Working Papers in Linguistics*, 8:128–172, 1986.

[45] M.F. Garrett. The organization of processing structure for language production. In D. Caplan, A.R. Lecours, and A. Smith, editors, *Biological Perspectives on Language*. MIT Press, Cambridge, Mass., 1984.

[46] Petra Geutner. Introducing linguistic constraints into statistical language modeling. In *Proceedings of the 1996 International Conference on Spoken Language Processing (ICSLP)*, Philadelphia, October 1996.

[47] E. M. Gold. Language identification in the limit. *Information Control*, 10:447–474, 1967.

[48] H. Goodglass. Studies on the grammar of aphasics. In H. Goodglass and S. Blumstein, editors, *Psycholinguistics and Aphasia*. Johns Hopkins University Press, Baltimore, 1973.

[49] Jane Grimshaw. Form, function, and the language acquisition device. In C. L. Baker and John J. McCarthy, editors, *The Logical Problem of LAnguage Acquisition*, pages 165–182. MIT Press, Cambridge, Mass., 1981.

[50] Y. Grodzinsky. The syntactic characterisation of agrammatism. *Cognition*, 16:99–120, 1984.

[51] P. Grünwald. A minimum description length approach to grammar inference. In Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 203–216. Springer-Verlag, Berlin, 1996.

[52] M. A. K. Halliday. *Language as social semiotic*. Edward Arnold Publishers Ltd., London, 1978.

[53] F. Harary. *Graph Theory*. Addison-Wesley, 1969.

[54] C. Heeschen. Agrammatism and paragrammatism: a fictitious opposition. In M. L. Kean, editor, *Agrammatism*. Academic Press, New York, 1985.

[55] X. Huang, F. Alleva, H. Hom, M. Hwang, K. Lee, and R. Rosenfeld. The Sphinx-II speech recognition system: An overview. *Computer, Speech and Language*, 2:137–148, 1993.

[56] R. Isotani and S. Sagayama. Speech recognition using particle n-grams and content-word n-grams. In *Proceedings of Eurospeech '93*, pages 1955–1958, Berlin, September 1993.

[57] F. Jelinek. Markov source modeling of text generation. In J. K. Skwirzinski, editor, *The Impact of Processing Techniques on Communications*. Nijhoff, Dordrecht, 1985.

[58] F. Jelinek. Up from trigrams. In A. Waibel and K. F. Lee, editors, *Readings in Speech Recognition*. Morgan Kaufman, San Mateo, California, 1990.

[59] F. Jelinek, J. D. Lafferty, and R. L. Mercer. Basic methods of probabilistic context-free grammars. In *Speech Recognition and Understanding: Recent Advances, Trends and Applications. Proceedings of the NATO Advanced Study Institute*, pages 345–360, 1992.

[60] S. Johansson, E. Atwell, R. Garside, and G. Leech. *The tagged LOB corpus*. Norwegian Computing Centre, Bergen, Norway, 1986.

[61] D. Johnson and P. M. Postal. *Arc Pair Grammar*. Princeton University Press, Princeton, 1980.

[62] M. Joos. *Readings in Linguistics*. American Council of Learned Societies, Washington, 1957.

[63] Martin Kay. Functional grammar. In Christina Chiarello et al., editor, *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society*, pages 142–158, 1979.

[64] Rick Kazman. Why do children say "me do it?". In K. J. Hammond and D. Gentner, editors, *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, pages 455–460, Hillsdale, New Jersey, 1991. Lawrence Erlbaum.

[65] Rick Kazman. Simulating the child's acquisition of the lexicon and syntax—experiences with *Babel. Machine Learning*, 16:87–120, 1994.

[66] M. L. Kean. The linguistic interpretation of aphasic syndromes: agrammatism in Broca's aphasia, an example. *Cognition*, 5:9–46, 1977.

[67] David Kirsch. PDP learnability and innate knowledge of language. *CRL Newsletter*, 6(3), December 1991.

[68] H. Kucera and W.N. Francis. *Computational analysis of present-day American English.* Brown University Press, Providence, Rhode Island, 1967.

[69] John D. Lafferty, Daniel Sleator, and Davy Temperley. Grammatical trigrams: a probabilistic model of link grammar. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 89–97, Cambridge, Mass., 1991.

[70] Howard Lasnik. *Minimalist analysis.* Blackwell Publishers, Malden, Mass., 1999.

[71] G. Leech, R. Garside, and M. Bryant. Claws4: The tagging of the British National Corpus. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94*, pages 622–628, Kyoto, Japan, 1994.

[72] Ming Li and Paul Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer Verlag, New York, second edition, 1997.

[73] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1):22–31, March 1968.

[74] J. Macnamara. *Names for Things: a Study of Child Language.* Bradford Books/MIT Press, Cambridge, Mass., 1982.

[75] D. Magerman and M. Marcus. Parsing a natural language using mutual information statistics. In *AAAI-90: Proceedings of the 8-th National Conference on Artificial Intelligence*, 1990.

[76] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, Mass., 1999.

[77] Andrei A. Markov. An example of statistical investigation in the text of 'Eugene Onyegin' illustrating coupling of 'tests' in chains. In *Proceedings of the Academy of Science, St. Petersburg*, pages 153–162, 1913. volume 7 of VI.

[78] James D. McCawley. *Thirty Million Theories of Grammar.* Croom Helm Ltd, London, 1982.

[79] L. E. McMahon, L. L. Cherry, and R. Morris. Unix time-sharing system: Statistical text processing. *Bell Sys. Tech. J.*, 57(6):2137–2154, 1978.

[80] G. Miceli, A. Mazzucchi, L. Menn, and H. Goodglass. Contrasting cases of Italian agrammatic aphasia without comprehension disorder. *Brain and Language*, 19:65–97, 1983.

[81] Richard Montague. Formal philosophy. In R. H. Thomason, editor, *Selected Papers of Richard Montague.* Yale University Press, New Haven, CT, 1974.

[82] S. Muggleton, A. Srinivasan, and M. Bain. Compression, significance and accuracy. In *Proceedings of the Ninth International Machine Learning Conference*, pages 338–347, San Mateo, CA, 1992. Morgan-Kaufmann.

[83] K. Nagita. Licensing functional categories. In *Proceedings of the Edinburgh Linguistics Department Conference*, pages 142–151, University of Edinburgh, 1994.

[84] T. R. Niesler and P. C. Woodland. Variable-length category-based n-grams for language modeling. Cued/f-infeng/tr.215, Cambridge University Engineering Department, Cambridge, April 1995.

[85] Steven Pinker. *Learnability and cognition.* The MIT Press, Cambridge, Mass, 1989.

[86] Steven Pinker. Why the child holded the baby rabbits: A case study in language acquisition. In L. R. Gleitman, Liberman M., and Osherson D. N., editors, *An Invitation to Cognitive Science, 2nd Edition, Volume 1: Language.* MIT Press, Cambridge, Mass., 1997.

[87] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[88] D. W. Quine. Comments on W. Newton Smith's *The underdetermination of theory by data. Proceedings of the Aristotelian Society Supplementary Volume*, 52:71–91, 1978.

[89] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:416–431, 1982.

[90] R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* PhD thesis, Carnegie Mellon University, 1994.

[91] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228, 1996.

[92] B. Y. Ryabko. Data compression by means of a 'book stack'. *Problemy Peredachi Informatsii*, 16(4), 1980.

[93] H. Schmid. Part of speech tagging with neural networks. In *Proceedings of the International conference on Computational Linguistics*, 1994.

[94] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:398–403, 1948.

[95] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

[96] B. F. Skinner. *Verbal behaviour.* Appleton-Century-Crofts, New York, 1957.

[97] Daniel Slater and Davy Temperley. Parsing English with a link grammar. Technical Report CMU-CS-91-196, Department of Computer Science, Carnegie Mellon, 1991.

[98] Tony C. Smith. Language inference from a closed-class vocabulary. Master's thesis, University of Calgary, Canada, March 1993.

[99] Tony C. Smith. Learning feature-value grammars from plain text. In David M. W. Powers, editor, *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Natural Language Learning*, pages 291–294, Adelaide, Australia, January 1998. working paper.

[100] Tony C. Smith. Learning inflection agreement from parental speech. In Eve V. Clark, editor, *Proceedings of the 30th Child Language Research Forum*. Stanford Center for the Study of Language & Information, Cambridge University Press, 1999.

[101] Tony C. Smith and John G. Cleary. Probabilistic unification grammars. In *Workshop Notes: ACSC '97 Australasian Natural Language Processing Summer Workshop*, pages 25–32, Macquarie University, February 1997.

[102] Tony C. Smith and Ross Peeters. Fast convergence with a greedy tag-phrase dictionary. In James A. Storer and Martin Cohn, editors, *Proceedings of the IEEE Data Compression Conference*, pages 33–42, Los Alamitos, California, March-April 1998. IEEE Computer Society.

[103] Tony C. Smith and Ian H. Witten. A genetic algorithm for the induction of natural language grammars. In *Proceedings of the IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing*, pages 17–24, Montreal, Canada, August 1995.

[104] Tony C. Smith and Ian H. Witten. Probability-driven lexical classification: A corpus-based approach. In *Proceedings of PACLING-95*, pages 271–283, University of Queensland,Brisbane, Australia, April 1995.

[105] Tony C. Smith and Ian H. Witten. Learning language using genetic algorithms. In Ellen Riloff Stefan Wermter and Gabriele Scheler, editors, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 132–145. Springer, Heidelberg, New York, 1996.

[106] Tony C. Smith, Ian H. Witten, John Cleary, and Shane Legg. Objective evaluation of inferred context-free grammars. In *Proceedings of the 1994 ANZIIS*, Brisbane, Australia, November 1994.

[107] R. J. Solomonoff. A formal theory of inductive inference, parts 1 and 2. *Information and Control*, 7:1–22,224–254, 1964.

[108] R. J. Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Trans. Information Theory*, IT-24(4):422–432, 1978.

[109] Margaret Speas. *Phrase structure in natural language.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.

[110] T. Stowell. *Origins of Phrase Structure.* PhD thesis, MIT, Cambridge, Mass, 1981.

[111] W. J. Teahan and John C. Cleary. Tag-based models of English text. In James A. Storer and Martin Cohn, editors, *Proceedings of the IEEE Data Compression Conference*, Los Alamitos, California, March-April 1998. IEEE Computer Society.

[112] T. A. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, June 1984.

[113] Jinxi Xu and W. Bruce Croft. Corpus-based stemming using co-occurrence of word variants. Technical report tr96-67, Dept. of Computer Science, University of Massachusetts/Amherst, January 1998.

[114] Deniz Yuret. *Discovery of Linguistic Relations Using Lexical Attraction.* PhD thesis, Massachussetts Institute of Technology, May 1998.

[115] George K. Zipf. *Human behaviour and the principle of least effort.* Addison-Wesley, Reading, Mass., 1949.