Statistical Language Processing based on Self-Organising Word Classification

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy
in the
Faculty of Science
of
The Queen's University of Belfast

By
John George Gavin McMahon, BA., MSc.
September 1994

Abstract

An automatic word classification system has been designed which processes word unigram and bigram frequency statistics extracted from a corpus of natural language utterances. The system implements a type of simulated annealing which employs an average class mutual information metric. Resulting classifications are hierarchical, allowing variable class granularity. Words are represented as *structural tags* — unique n-bit numbers the most significant bit-patterns of which incorporate class information. Therefore, access to a structural tag immediately provides access to all classification levels for the corresponding word. The classification system has successfully revealed some of the structure of two natural languages, from the phonemic to the semantic level. The system has been favourably compared — directly and indirectly — with other word classification systems. Class based interpolated language models have been constructed to exploit the extra information supplied by structural tag classifications. Experiments measuring test set perplexity have shown that the new models significantly improve performance.

Contents

List of Figures

List of Tables

Firstly I would like to express thanks to my supervisor, Professor F. J. Smith, for providing me with the oppurtunity to work on statistical language modelling at Queen's and for his conscientious and useful comments on earlier drafts of this thesis. Secondly thanks to my unofficial second supervisor, Dr. Peter O'Boyle, who has helped me to improve my grasp of Markov models, interpolation, weighted average language models, probability theory and statistics. Also, his technical help has been invaluable. I would also like to thank several other people for their technical assistance : Jenny Johnston, Stephen Fitzpatrick, Tony McHale, Jaron Collis and David Gault. The computer science secretaries Lynn, Janet and Helen have also been helpful. Thanks to Pat Crookes for his knowledge of Latin.

The following people have made the last three years at Queen's more interesting and enjoyable : Adam Winstanley, Marie Owens, Paul Donnely, Paul G. Donnelly, Bernie Stuart, Neville Sinnamon, Philip Crooks, Krishna, Shakti, Adib and Moira Watson; Keith McVeigh, William Crawley, John Catherwood and the rest of the Philosophy Department Seminar Group.

British Telecom Research Laboratories supported this work with a CAST award. I would like to thank some people working in the Speech and Language Technology Group at British Telecom's Research centre near Ipswich for making my stays bearable : Peter Wyard, William Millar, Mark, Dean, John, Sandra, Geoff, Ambi, Carlos, Patrick, Owen, Norbert, Maria and Esther.

This work was also supported in part by a grant from the Department of Education for Northern Ireland.

British Telecom supplied the VODIS corpus; all other corpora were supplied by the Oxford Text Archive.

Finally, thanks to my partner Sharon.

Chapter 1

Language Processing — An Introduction

1.1 Models of Language

The construction of models of the human language processing capacity is a useful research activity; not only can these models shed light on the mechanisms by which humans acquire and use language, but they can lead to impressive and practical engineering systems in their own right (for example, in speech recognition, optical character recognition, automated proof reading, lexicographical support tools, automatic translation and information retrieval). These two perspectives are each adequately represented in current research activity.

Humans generate and recognise language, but many current language models embody a specialism in recognition. The reason is that coherent all-encompassing theories of the process of language generation are rare, and in all likelihood, more complicated than recognition theories. The models of language described in this thesis are recognition based; they are designed as finite state grammars, which occupy a lowly position in the Chomsky hierarchy. One of Chomsky's many contributions to linguistics was a series of arguments against the sufficiency of finite state grammars as models of the human language processing ability. In the years following the first elaboration of these powerful and convincing arguments, little effort was expended on investigating finite state grammars as language models. Then, as computers became more powerful, as large natural language corpora became widely available and as the application of finite state machines began to show promise in the related research domain of speech recognition, some language modellers turned with renewed interest to finitary models. It is as part of this research tradition that the current thesis belongs.

We accept many of the criticisms of finitary models but still make use of them due to their

ease of construction, the minimal degree of linguistic assumption they entail, their robustness and their wide utterance coverage when used as recognition aids.

1.2 Refining Finitary Models

Language model performance can be reasonably estimated; this allows us to identify and pursue research directions which might result in the development of more useful language models. This is precisely the goal of the work described in this thesis. It should be emphasised that the statistical approach to language processing is independent of the type of grammar being used — see, for example, the work of Salomaa [138] and Lari and Young [97] who discuss stochastic context-free grammars as language models. However, finitary models are theoretically close to a family of stochastic models usually described as Markovian; the mathematics of these models has been well described in the literature and so it has been easy for language modellers who are interested in the probabilistic nature of language to exploit finitary models first.

We shall be exploring finitary models which use information about word classes. Some work has been carried out in this area before, but most of it makes use of pre-tagged corpora or lexica. Our models will use class information which has been extracted automatically from untagged corpora. To do this, we shall build an automatic word classification system.

1.3 Automatic Word Classification

Work on building automatic or semi-automatic word classifiers can — and indeed often does — proceed independently of work on improving finitary recognition models. However, there are deep connections : having information about the class of a word reduces the difficulty of predicting which word is likely to come next in a stream of words; conversely automatic word classification works by noting which words typically follow other words in a stream — that is, which words are typically good predictors for other words. One of the advantages of considering these two areas as autonomous is that automatic word classification researchers do not have to commit themselves to accept that finitary models are cognitively plausible. It is entirely possible to maintain a transformational position on syntax, for example, and still offer an information-theoretic explanation of vocabulary and word-class acquisition; the universal grammar which Chomsky describes is vocabulary independent and word-class independent.

We present a new automatic word classification system which implements a type of simulated annealing; we also introduce a word-classification representation which has been de-

signed to be incorporated into class based statistical language models.

1.4 Finitary Models as Cognitive Models

A discussion on the merits and limitations of finitary models of language is postponed until the final chapter, but we shall make some general observations now, since it may be argued that finitary models, being demonstrably inadequate from a theoretical linguistic perspective (Pinker in [124] has recently restated this claim) are therefore not worthy research topics (Ramsay in [130] has drawn this conclusion for finitary models in cognitive science).

Computational models of language are usually implemented on computers, which are finite state machines. It has been argued that finitary models are *in principle* inadequate due to their finite nature. This argument, however, also holds for all implemented language models since unlimited recursion is not possible using computers. In other words, a piece of software which simulates, for example, a context sensitive grammar can be regarded as a finitary model, and so it cannot properly handle unbounded dependencies. For every implemented model of a context-free or context-sensitive grammar a weakly equivalent finite-state grammar can be constructed simply by analysing the micro-instructions of the computer system. If it can be shown that humans are also finite state machines, then they too cannot handle unbounded dependencies. Chomsky claims that these human limitations, being psychological in nature, impact on the performance characteristics of language; the underlying competence model is describable in a hardware-independent way. But that humans are embodied and have language performance limitations is an important aspect of human language processing; consequently, good language models must incorporate this fact. In software engineering, by analogy, assuming that a computer is Turing equivalent, or that two algorithms are weakly identical (in the sense that they have exactly the same input-output behaviour) does not help builders of practical systems, since, for example, any sorting algorithm would be as valid as any other. Instead, concepts such as algorithmic complexity and time efficiency are important. Some computational linguists make similar points : a language model which is incapable of processing a sentence which has several hundred levels of embedding might not be considered too inadequate if it can recognise many well-attested sentences.

Given that all (implemented) computational models of language are describable as finitary, at a low functional level, it still may be the case that systems which, at higher functional levels of description, operate in non-finitary ways may be better language models. This could be because they are better at recognising sentences, or better at generating them, or make errors which correspond to the sorts of mistake humans often make, or, when perturbed at certain

places, exhibit pathologies which are also like those witnessed in humans who have damage to areas of their brain which have equivalent functionality. When measured against these criteria, we are forced to conclude that the sorts of language model which will be described in the current thesis are inadequate as cognitive models of the human language ability. We do not claim the same conclusions, however, for all finitary models; indeed, we suggest that the insights afforded by the current spate of activity with explicitly finitary language models — their propitious use of the stochastic paradigm, the increasingly clear definition of the boundaries of their power, their wide-coverage and their application to the processing of the sorts of utterance which humans regularly make — provide a useful platform from which to construct genuinely insightful cognitive models and eminently useful engineering applications.

We can illustrate the nature of the inadequacy of explicitly finitary models of language by analogy to an aeronautical engineer who, upon the design and successful testing of a jet-propelled aeroplane, proceeds to suggest that birds use jet propulsion to fly. The engineer has constructed a valuable flying system, but a wholly unsatisfying model of bird flight. With automatic word classification, however, the case for making slightly stronger — though still not very strong — claims about cognitive plausibility is discussed in the final chapter.

1.5 Background theory

1.5.1 Language as a sign system

All of the research of this thesis involves a presumption that language is a sign system [40]. This approach casts language as an abstract structured object whose primitive atom is the *sign*; this basic unit has two indissoluble characteristics — the signifier and the signified. The signifier can be seen as the prototypical 'sound image' of the sign and the signified as the concept represented or meant by the signifier. The connection between the signifier and the signified is essentially arbitrary — all natural languages are the product of convention. So, in English, for example, the word ⟨`book`⟩ and the concept of a book have no necessary connections. What, then, is the criterion of identity for signs of particular sign systems? Saussure claimed that this was determined by a sign's relations to all of the other signs in the system. The 'value ' of a sign is its place in the system — its structural role. This perspective on language has been assimilated by researchers in all areas concerned with language (see [141, 69, 75, 159, 91, 45, 106] for work which incorporates some structuralist perspectives into linguistics, anthropology and cognitive science, and see [82, 154] for elaborations of structuralism into sociolinguistics and psycholinguistics).

The usual example of signifiers are word utterances — but others can exist : certain hand and body movements, associated with facial expressions can constitute signifiers, as can written text, a particular example of which is the ASCII encoding of letters in conventional computer-readable natural language corpora.

Saussure starts his analysis of the linguistic object with the introduction of two special relations which describe a sign's overall position in the structure. These are *syntagmatic* and *paradigmatic* relations. The former govern internal and external rules of combination of units (*e.g.* sentences, well-formed formulae, morphology, phrases) and the latter hold among sets of signs which have some feature in common — *i.e.* some syntactic similarity (⟨`book`⟩, ⟨`cat`⟩ and ⟨`table`⟩ belong to a set of words which have a certain lexical distribution; linguists usually refer to these words as nouns), or a semantic similarity (⟨`cat`⟩ and ⟨`dog`⟩ belong to the set of animate objects whereas ⟨`table`⟩ and ⟨`book`⟩ belong to the mutually exclusive set of inanimate objects). The paradigmatic relation can be seen as a move upwards in abstraction in the process of describing the structure of the linguistic object. At any level, the syntagmatic relation describes the structure of particular sentence instances, or, in conjunction with the paradigmatic relation, the structure of classes of sentence instances.

1.5.2 Language as more than a sign system

There is another, equally fundamental approach to the study of language which views it as the common property of groups of individuals, transmitted culturally — a system of situated and related behaviours (see [158, 104, 68] for philosophical, non-linear dynamic and computational linguistic perspectives). The main difference underlying these two approaches lies in the varying breadths of understanding of our idea of context.

The narrow, well-defined sense of context is linguistic context — the words which precede and follow a given word. A richer definition [113] includes more sociolinguistic and psycholinguistic characterisations : who said what to whom, and in what cultural situation, for what purpose? Experiential backgrounds of participants and current moods and conversation topics become important considerations.

Many researchers in the field of language processing have thus far usually assumed that considering language as a disembodied sign system is sufficient, partly due to a realisation that they are inadequately equipped to deal with a full treatment of context; it is also partly due to the success of Chomsky's arguments about the autonomy of syntax.

Saussure's relational approach may not be enough to fully characterise language phenomena, but it is still a potentially powerful tool for analysis of given sign systems. If the

structural approach is used as a method of description rather than a device for the production of language — that is, as a recogniser rather than as a generator — then some of these criticisms can be avoided. Ultimately, though, the problem of a sign system's connections with use and context cannot be avoided in works which aim to investigate natural language phenomena fully.

A realisation of the scope of extra-linguistic context usually results in investigators excluding it from consideration, but one can appreciate its enormity yet resolve to concentrate on well-defined areas, exploring at least some of these extra-linguistic contextual cues. This must be recognised from the outset to be artificial and incomplete — no set of cultural situations is hermetically sealed : anthropologists realsie this, but still resolve to do fieldwork. However, in the field of computational linguistics, situated language understanding systems are very rare; Gorin *et al.* [68] describe one of the better examples.

1.5.3 Corpus and Language : Defining the differences in the field of possible objects of study

The various schools of thought in language processing often deal with different linguistic phenomena. One major axis of difference is between sample/corpus and the total set of well formed formulae of the sign system. Church and Gale [31], for example, take as their language universe only those utterances found on the Associated Press newswire. The danger with any sample-based work is that the results concluded from such samples do not scale up well to full languages. Chapters 2 and 3 provide some evidence for this concern. If the sample happens to originate from a particular domain, however, then the utterances of that corpus are easier to predict.

With broad-coverage corpora, there are additional complications when considering this 'scale-up' problem, which are connected to issues in cognitive anthropology. Natural languages are situated in social contexts; for any given culture which is said to share a natural language, there are many dialects and other non-dialectical variations which result in differing pockets of language behaviour. Just as in anthropology it has been realised that, for a culture, there is no *one* ideal informant who appreciates every cultural nuance, so too it must be made clear that the extension of a language model from a corpus to a complete sign system will be too fragile a transition to make without undesirable over-simplification, especially when we consider that the composition of existing corpora may be too fractured and homogeneous. This insight, if properly understood, could lead to a re-appraisal of the role of corpora.

Linguists and cognitive scientists have thus far generally considered natural language as that abstract object which is manipulated by the ideal human being. In contrast, our notion of culture is as a pool of knowledge from which each person takes a small part — at different places depending upon their age, sex, social position, *etc.* and by differing amounts, depending on skill, experience, intelligence, social status and many other factors. The cultural inhabitant can take on a number of roles at different periods in their lifetime, at different geographical locations, in different social situations; for any given social situation, culture is defined as that knowledge which is necessary for an agent to act correctly (*i.e.* in accord with the appropriate social norms for the situation). This is true of linguistic behaviour also; the knowledge of English is distributed amongst members of all of the English speaking cultures.

Modern anthropologists recognise that much time and effort needs to be spent on relatively small aspects of cultural behaviour. They hope that through this cumulative ethnographic approach, a fuller description of a given culture will emerge.

There are many forms of human behaviour which are highly formalised and narrowly circumscribed. For example, religious behaviour in many cultures observes well-described patterns. Anthropologists can discover the structure of these behaviours. Again, parts of these behaviours involve language in greater or lesser degrees of regularisation. Often in religious rituals there is a highly constrained dialogue between religious leader and a member or members of the community. There is no doubt that the participants of any such scene also involve themselves in other, often more complex roles; but as worshipper at a ritual, their behaviour is more formalised. Anthropologists have traditionally concentrated on those dimensions of cultural life more amenable to structural analysis — religion, kinship and political organisation. So too for a researcher interested in communication through natural language. The VODIS corpus [36] provides a useful example for us; it is a small corpus — less than 100,000 words — which contains transcriptions of telephone enquiries made to British Rail in Ipswich about the train time-table. The conversations occurred in 1985. Recorded telephone conversations are useful because the physical aspect of this particular set of cultural interactions can be neglected. Also, while there is no doubt that the caller and the receptionist both have much more complicated linguistic repertoires, but for the purpose of describing the structure of these verbal encounters, we only need consider the caller in their role as caller and the receptionist as information provider.

The most obvious overarching structural description of these interactions is as a particular type of dialogue concerned mainly with propositional questioning and answering. Thus the caller — or questioner — is not in general interested for example, in the emotional state of

the receptionist — or answerer. Indeed the propositional structure of the questions is itself highly constrained — which amounts to saying that this set of interactions are themselves embedded in a social context. The telephone number which the questioner dials is advertised as a British Rail Enquiries number. Both parties share in common many cultural facts — a general knowledge of trains, timetables, money and geography; of how to engage in dialogue over telephones; of how to ask and answer questions. Even the fact that the office number is an Ipswich one should lead to a certain structural pattern — the destinations and points of departure are more likely to be local than distant; certain place names tend to come up in conversation more than others. This can be a help in processing ambiguous utterances and it also eases the flow of dialogue.

The answerer possesses more knowledge about the domain than the questioner, but the questioner possesses the problem-specific knowledge which determines answers. Also, the experienced answerer begins to get a feel for a taxonomy of questions — this helps them arrive at an understanding of what information is required and leads to a more efficient exchange. These factors are, of course, dependent upon the communicative skills and the experience of both participants (particularly the answerer). A dialogue structure can emerge — involving typical questions and typical answers. This structure is reflected in part in the lexical and syntactic details of the corpus. Stephens and Beattie [151] investigate part of this structure; also, table 4.2 and figure 5.11 show some of this structure for the VODIS corpus. In anthropology, this type of approach is called ethnomethodology [90, 72, 82].

The question of whether a corpus of a certain size (for example, the VODIS corpus) can be extended to be statistically relevant to the language of which it is a part can thus be countered in the short term by suggesting that, like anthropologists, all that current workers in this area can do is adopt a piecemeal methodology and add that the kinds of scale-up that they find interesting are not of a vague, generic kind, but of a quite specific, situationally related kind. In other words, we may be less interested in English language conclusions drawn from a VODIS corpus sample and more interested in

- restricted domain question asking and answering and the linguistic behaviour thereof
- Ipswich-centred British Rail enquiries and related speech acts

Also, although there are infinitely many ways of linguistically structuring a finite set of questions, there is an assumption that most questions are formulated in a restricted, culturally defined number of ways. The more specific one's view of culture — that is, the more accurate it is — the more linguistics is replaced by a study of dialects, and then of idiolects : 'Language'

becomes the composite of the languages of the plumber, the anatomist, the chemist and the politician. Sociolinguists and cognitive anthropologists are more interested in the regularly used linguistic forms situated in cultural interactions than the set of all well-formed formulae of a sign system.

1.6 Thesis Structure

This section summarises the contents of the remaining chapters. Chapter 2 introduces the most successful finitary models of language — those based on Markov model theory. Here, there are states, which correspond to a finite word-context memory, and transition probabilities, which correspond to conditional probabilities of words. These models are also referred to as n-gram models of language. We introduce the most useful of the models reported in the literature. Then we address the problem of the sparseness of frequency data extracted from presently available corpora. Next, we summarise the main ways in which this problem has been confronted. At this stage, we introduce the measure of perplexity as a way of evaluating language models. Then, some technical issues relating to the format of corpora are raised, together with an experiment which illustrates the importance of formatting, and which demonstrates the use of the perplexity measure. Finally, a short summary is given of the stages involved in the construction of an n-gram frequency count database.

Having been introduced to the main techniques of n-gram statistical language modelling, in chapter 3 we show how they can be extended to include class information. We subsume word-based language models within a broader representational framework and describe the four main types of class-based database. By way of further introduction to the field, a simple classification system is used to tag word types. The resulting class information can be used to explore the possible advantages which classes offer language modellers. Several experiments follow which show that the type of class database used, together with the type of classification system, are essential to the quality of any resulting language model. Next, we observe that using class information requires us to make sensible estimates of the conditional probability of a word given a class. An experiment lends support to this hypothesis; it also provides an introduction to the design of class-based language models. Some observations about the results from several related language model experiments lead to the introduction of a new classification representation — the structural tag; also, a way of incorporating structural tags into language models is suggested.

The structural tag representation naturally leads us to consider types of classification system, and to speculate about the possibility of deriving a classification automatically. To

this end, chapter 4 offers an overview of some automatic methods in language processing.

Chapter 5 introduces a new algorithm which manipulates structural tag representations to produce automatic word classes. Then, several experiments are performed, on various corpora, to identify the range and depth of performance of the algorithm : it succeeds in revealing some phonemic, syntactic and semantic structure of English. It also reveals some of the structure of Latin.

Next, chapter 6 attempts to make a comparative evaluation of the system with the best-known word classification systems. We include a discussion on the difficulties of fair comparison in this chapter. We also implement another type of classification system, which is based on word merging and we directly compare results from this implementation with our own. Then we implement a benchmark classification system, described in the literature, in order to further evaluate our system. After the main evaluations, we discuss some of the strengths and limitations of our system.

In chapter 7, we incorporate the newly available class information into class-based language models. To this end, we make use of a parameter estimation technique common in Markov model theory. We also implement some word-based language models and use their test set perplexities as points of reference for results from the class-based systems. Before we can construct these models, we introduce a new supplementary classification algorithm, which is less optimal, but which allows greater vocabulary coverage. The new hybrid system can classify tens of thousands of words. Finally, some class-based language models are implemented and their perplexities reported.

Finally, chapter 8 summarises the main results of this thesis, emphasising the significant contributions as well as the limitations. It also relates the present work to some issues in the cognitive science of natural language processing and includes suggestions for further work.

Chapter 2

Word Based Language Models

2.1 Overview

This chapter provides an introduction to the key terms and concepts of statistical language modelling. It starts by describing the main source of the statistics, the corpus itself; next it describes the function of all language models, namely to estimate the likelihood of word sequences [85, 5]. Then, the benefits of decomposing word sequences are discussed.

After that, the nature of the sparseness of frequency statistics derived from corpora is introduced [119], and how the distribution of frequencies varies with word sequence length. This empirically discovered fact provides a challenge for researchers to exploit to its full potential that region of the space of word sequence frequencies where statistically useful information resides [85]. A simple trigram language model is introduced to demonstrate an immediately obvious way of exploiting information-rich word sequences.

A natural extension of this discussion is to investigate the scope and limitations of moving beyond trigram dependent models and into n-gram models [18, 85]. After this, the technique of combining two or more functioning language models is introduced. Several model combinations from the literature are re-described from a more general perspective. The relative merits of these various ways of combining simpler models are discussed.

Although it is often qualitatively clear that some models or model combinations are better than others, a more quantitative measure of language model performance is introduced next. This is *test set perplexity* [114, 86, 85].

By this stage, we will have discussed what language models are, how they are built and how they are evaluated. The next section deals with some of the issues which need to be confronted by language model researchers when they attempt to implement their own language models. Results of an introductory experiment which shows how the format of a

corpus influences the test set perplexity are discussed.

Finally, a brief summary is given and the possible role of word classes in language modelling is introduced.

2.2 Compositionality

A corpus can be described as $C = \langle s_1, s_2, \ldots, s_i, \ldots, s_N \rangle$ where s_i is the ith element of the ordered list of N elements. Initially it can be assumed that the elements are words [86] (defined simply in terms of their occurrence between separator characters). Frequency statistics can be extracted from C and used to calculate the estimated probability of any sequence of words; the generation of the raw frequencies corresponds to building a language model and the estimation of word sequence probabilities corresponds to using that model. All language models, then, are devices which take as input a word sequence and produce as output a probability score [84, 85, 86].

These language models are considered useful whenever they output high probabilities for word segments which appear frequently in C and low probabilities for those segments which are unlikely to occur in C. If C contains a sufficiently representative sample of utterances from the underlying language, then the estimated probabilities are claimed to be applicable to the language itself. In information theoretic terms, this is equivalent to saying that a good language model is one which has low uncertainty about input samples of language which are representative and high uncertainty about unrepresentative samples.

The goal of developing a model which produces a useful probability,

$$\hat{P}(\langle w_1, w_2, \ldots, w_n \rangle)$$

where $\langle w_1, w_2, \ldots, w_n \rangle$ is a sequence of n words, is advanced by recognising the following important principle of compositionality [84], which holds for any probabilistic sequence of events, including the sequence of words which constitutes C :

$$\hat{P}(\langle w_1, w_2, \ldots, w_n \rangle) = \prod_{i=1}^{n} \hat{P}(\langle w_1, w_2, \ldots, w_i \rangle | \langle w_1, w_2, \ldots, w_{i-1} \rangle) \tag{2.1}$$

where the component $P(w_1|w_0)$ is fixed as the unconditional unigram probability of w_1.

Equation 2.1 allows us to estimate the required probability as the product of a series of conditional probabilities. This is particularly useful for language model researchers for at least the following reason. The equation suggests how this probability estimation can be implemented as an iterative process, the heart of which involves the estimation of the

conditional probability $\hat{P}(\langle w_1, w_2, \ldots, w_i\rangle|\langle w_1, w_2, \ldots w_{i-1}\rangle)$; this can also be expressed in the more compact notation $\hat{P}(w_1^i|w_1^{i-1})$. A conditional probability unit like the one above exists for each new word in the input segment W. In effect, this conditional probability estimates how likely w_i is to be the next word. The model functions as a finite order Markov Model (see section 7.2 for a more detailed discussion).

The most obvious application of statistical language modelling is as a post-acoustic component of a speech recognition system, or in an integrated speech system where word segment probabilities effect ongoing hypotheses about the acoustic signal [83, 120]. A speech recogniser is a system which maximises the probability $P(S|A)$ of some segment, S having been the cause of the observed acoustic signal A. Bayes' formula allows us to re-write the probability as :

$$P(S|A) = \frac{P(A|S) \times P(S)}{P(A)} \tag{2.2}$$

For a given input signal, the denominator in the above fraction remains constant; of the two elements of the numerator, $P(A|S)$ is known as the acoustic model — it estimates the likelihood of a signal being observed given prior knowledge that word segment S has been uttered — and $P(S)$ is the language model, which is an estimate of the prior probability of a segment. The importance of the language model component can be seen by considering the following example. If the speaker says

```
The boys eat the sandwiches.
```

and we assume that the acoustic model is perfect, it will therefore be able to recover the optimal phoneme string :

```
/DH a b OI z EE t DH A s AA n d w i j i z/
```

However, the original sentence is not the only speech utterance which could give rise to the observed phoneme string. For example, the meaningless and ungrammatical sentence

```
The buoy seat this and which is.
```

also gives rise to the observed phonemic stream. Such sound streams are known as oronyms. Humans re-construct the most likely sentence very successfully, but speech recognisers with no language model component cannot. A useful statistical language model will assign a low probability to the second sentence and a high probability to the first. A more traditional, non-probabilistic language model, in the form of a grammar, could also differentiate between the two sentences, weeding out the second, ungrammatical sentence. However, such models, whilst theoretically well grounded, so far tend to have poor coverage. Another problem with non-probabilistic models can be seen if we consider a third hypothesised sentence :

`The buoys eat the sand which is.`

This simultaneously surreal and metaphysical sentence is accepted by grammar systems which only look for well-formedness, but it is subsequently indistinguishable from the correct hypothesised sentence. A probabilistic language model should assign a very low probability to the third sentence. This example also highlights one problem faced by language modellers who hope to use domain information to disambiguate sentences or to improve recognition performance : in the case of this sentence, some semantic priming might occur between the two content nouns of the hypothesised sentence : ⟨`buoys`⟩ and ⟨`sand`⟩ are both things found near the sea-side, but in this case, the semantic priming is unwarranted by the selectional restriction associated with the subject of the verb ⟨`to eat`⟩.

2.3 Sparse Data

The so-called sparse data problem [44, 58, 67] can be thought of as a technical one : given the size and power of current computers, no corpus can be processed which satisfies the requirement that it be sufficiently representative of the underlying language of which it is supposed to be an example.

There is also a theorerical dimension to the problem. It is argued that no corpus *could* provide enough frequency information to generate a model which fully captures the properties of the underlying language; for most segments in C, their frequency of occurrence is 1; also, a large number of segments are attested by the language, but which do not occur in C — *i.e.* their frequency of occurrence in C is 0.

Researchers in speech and language processing have tended to avoid letting this theoretical limitation restrict them and have concentrated their efforts on casting the problem as a technical one and attempting to find solutions to it [89, 95, 85, 31].

One of the most important structural features of a corpus of natural language concerns the relationship between the frequency of occurrence of a particular word segment and the length of that segment. In a corpus of N words, there are N sets of segments, each set containing segments of the same length. The sets of one-word segments, two-word segments and three-word segments contain most of the statistically useful information — that is, a significant proportion of segments of these sizes occurs with frequency greater than 1. As the word length increases, the corresponding segment frequencies tend towards 1.

It is important to realise that this feature of natural languages represents an interesting structural fact about language, not the mere unveiling of a problem. This fact began to be

thought of as a problem almost immediately however, because before its discovery it was hoped that there would be word segments in sufficient quantities in C to make language modelling straightforward.

If the word segments of language had been distributed less sparsely, then the conditional probability estimation, $\hat{P}(w_1^i|w_1^{i-1})$ could be estimated by $\frac{f(w_1^i)}{f(w_1^{i-1})}$. This probability estimation is called the maximum likelihood estimate. If both phrases only occur with frequency 1, which tends to happen for most segments, then $\hat{P}(w_1^i|w_1^{i-1})$ tends to 1, which results in a poor language model.

Given that the word segments of language are distributed sparsely, the method of language modelling could most usefully advance by paying special attention to those segments which occur frequently, namely the one, two and three word segments — often referred to in the literature as unigrams, bigrams and trigrams [73, 145, 85] — and, optionally, any longer segment which occurs greater than a specified minimum number of times.

Church and Gale [31] suggest that, even for a bigram language model, where there are in theory $V \times V$ parameters to be estimated (where V is vocabulary size), $V > O(\sqrt{N})$, where N is the size of the corpus; in other words, as corpus size increases, there will always be even more new bigram frequency parameters which need to be estimated and consequently the sparse data problem gets worse instead of better. This appears to be supported by Sampson [139], who notes that the number of rare noun phrase types in a moderate sample of authentic natural language utterances stands at just under two thirds of the total number of noun types; he further hypothesises that this relation will hold for larger text samples. Sampson uses this argument to strengthen the case for denying a grammatical/ungrammatical distinction, the opposite of which modern linguists often regard as axiomatic.

The probability decomposition described in equation 2.1 facilitates emphasis on small length segments as follows. In the conditional probability $\hat{P}(w_1^i|w_1^{i-1})$, the word sequence w_1^{i-1} can be thought of as representing a context for word w_i. It is a leftwards context, corresponding to those words which have already been spoken or read or recognised. For values of i larger than 3 or 4, $f(w_1^{i-1})$ will be close to 1 for most corpora in existence today. The full context w_1^{i-1} can be replaced by a smaller context which only includes the most recent two or three words. Thus $\hat{P}(w_1^i|w_1^{i-1})$ can be replaced, for example, by two-word recent histories : $\hat{P}(w_{i-2}^i|w_{i-2}^{i-1})$. If this is inserted into equation 2.1, we get the following *trigram language model* :

$$\hat{P}(w_1^n) = \prod_{i=1}^{n} \hat{P}(w_{i-2}^i|w_{i-2}^{i-1}) \tag{2.3}$$

This model can be thought of as a second order Markov model (see chapter 7 for a further discussion).

2.4 Beyond Trigrams

Many language models only deal with segments of three or less words [19]. Although this captures the bulk of segments which occur more than once, it is useful to have a combined language model which can take advantage of segments of any length which occur with a frequency greater than some specified minimal threshold. In order to achieve this goal, a filter mechanism needs to be applied which consists of those segments from the corpus which occur above the given threshold.

Without this filter, a practical limit usually has to be set so that only unigrams, bigrams and trigrams are used, for example. If there are V vocabulary items which the system recognises then a V^3 data array will be needed to store all of the possible trigram frequencies; not only is this infeasible for even moderately sized vocabularies, but it is also wasteful, due to the sparseness of the data. An improvement on this approach to trigram storage involves setting aside space only for those trigrams which occur in C — this is the simplest and most obvious filter.

A second filter, the frequency threshold filter, allows word segments of any length to be included, provided they occur with high enough frequency. This results in comparable language model performance and less training [119].

There is also a linguistic argument for considering wider contexts — phenomena like long distance dependencies require agreement between word forms but trigram models do not possess enough information to recognise these agreement patterns. Chomsky [27] criticises all finite order Markov models for not being able to deal fully with these phenomena.

2.5 Combining Language Models

Language models similar to that described by equation 2.3 can be constructed from two word and one word segment frequencies. Even better probability estimates can be achieved by combining two or more language models [31]. If equation 2.3 represents one way of approximating $\hat{P}(w_1^n)$, and the equivalent bigram and unigram models represent two other ways of approximating it, then the three models could be weighted together to produce a synthesised language model whose performance is measurably better. A general equation for describing how language models can be combined is :

$$\hat{P}(w_1^n) = \prod_{i=1}^{n} \sum_j \lambda_j \times \hat{P}_j(w_1^i | w_1^{i-1}) \tag{2.4}$$

where the λ_j values are weights and

$$\sum_j \lambda_j = 1$$

Several techniques are available for estimating how much weight to give each of the j language models.

2.5.1 Using a Held-Out Corpus

A sample of natural language which was not involved in the construction of the language models, and which will not be involved in any later testing of the synthesised model is used to maximise the probability score which the synthesised model estimates for that sample [85, 6] and see also chapter 7. That is, working on the assumption that the held-out sample is typical, we select that language model which gives us the highest corresponding probability [86, 84].

Each λ_j weight can be a simple constant, or it can be a function of some relevant feature of the language model sub-unit it weights. For example, the weight can be a function of the frequency of the previously processed segment or it can be a function of the syntactic or semantic class of the segment, or some combination of both. Generally, if the context of the jth language model is σ and $\phi(\sigma)$ is some partitioning function of the contexts then the weight associated with the jth language model is $\lambda_j(\phi(\sigma))$, whose sum over all j values is 1.

2.5.2 Manual Construction of a Weighting Function

Another approach is to use some of the insights gained from a knowledge of the structural properties of corpora to design and empirically test an explicit weighting function [119]. This method represents a move away from the approach which uses just those weight values which get the highest overall probability estimates, to one which concerns itself with more theoretical questions : what function best approximates the weight distributions found to be optimal using held out data? Why should this mathematical relation hold? In short, this method brings us closer to more descriptive analyses of the structure of corpora and, by implication, of language.

2.5.3 Conditional Use of Language Model Parts

Instead of trying to find optimal ways of incorporating each language model part, another approach involves using one model if and only if a certain set of conditions are met, another

model if and only if a second set of conditions are met, and so on. An example of this is a synthesised language model which uses the so-called *back-off* technique [84]. Here, λ_j is set to 1 or 0, depending upon whether or not the corresponding conditions are met. These conditions usually depend upon the frequency of a word segment. Thus, if the first language model consists of maximum likelihood unigrams, the second of bigrams and the third of trigrams, then if the raw trigram frequency is higher than some threshold, λ_3 is set to 1 and the other two weights are set to 0. If the trigram frequency is below its threshold but the bigram frequency is above its threshold, then λ_2 is set to 1 and the others are clamped to 0 again. Finally, if neither the trigram nor the bigram frequencies are over their thresholds, the synthesised language model reduces to a unigram language model — λ_1 being set to 1 and the rest to 0.

This set of combinations of language models is a subset of the range of possible combined language models described by equation 2.4, with simplifying constraints. Its main advantage is that it is easier to implement than a fully interpolated language model; its main disadvantage is that it does not perform as well.

2.6 Evaluating Language Models

In order to quantify the degree of utility of a model, a measure is made of its *perplexity*. Informally, perplexity (PP) is maximised in a system with a minimally informative language model. If language utterances are considered to proceed from a source and if there are L possible labels which could be emitted at a particular point, all of which are equiprobable and independent of previous emissions, then the value of PP (its maximal value in this case) is L. The constant lowering of PP corresponds to the generation of language models which are more useful. Thus the goal of improvement of language models could be achieved by minimising PP (the cited research uses this approach : [86, 84, 5, 19, 26, 18, 85, 89, 114, 144, 145]).

2.6.1 Formal Definition of Perplexity

A result of information theory [19] states that if a source can emit L labels with varying independent probabilities, then the *entropy*, H of the source — a measure of the uncertainty concerning the prediction of the next symbol — is

$$H = -\sum_{w=1}^{L} P(w) \log_b P(w) \tag{2.5}$$

where the base, b whilst theoretically irrelevant, is generally set to 2, so that H is measured in bits of information. All subsequent logs are assumed to be base 2 unless otherwise indicated. If every word w is equiprobable, then $P(w) = 1/L$ and equation 2.5 implies

$$H = \log L \tag{2.6}$$

If perplexity is defined as b^H, then we arrive at our original intuitive definition of perplexity (PP) as the branching factor of the source. If logs have been taken to base 2, then the perplexity is 2^H. Perplexity is independent of the base of logs and has been preferred by statistical language modellers.

Equation 2.5 can be extended to cover a sequence of words w_1^n. If S is the set of all sequences of n words, then

$$H = -(\frac{1}{n})\{\sum_{w_1^n \in S} P(w_1^n) \log P(w_1^n)\} \tag{2.7}$$

As n approaches infinity, the source entropy approximates the entropy of the underlying language. That is

$$H = -\lim_{n\to\infty} (\frac{1}{n})\{\sum_{w_1^n \in S} P(w_1^n) \log P(w_1^n)\} \tag{2.8}$$

If we assume that $P(w_1^n)$ reduces to $P(w_1)P(w_2)\cdots P(w_n)$ — *i.e.* that the probability of each word is independent of previous words, then equation 2.7 reduces to equation 2.5.

Also, if the source is *ergodic* — in other words, if sequences of words which are sufficiently long can be said to be typical of the output behaviour of the source and hence can be used to estimate the statistical structure of the source — then equation 2.8 gives us

$$H = -\lim_{n\to\infty} (\frac{1}{n}) \log P(w_1^n) \tag{2.9}$$

In practice, equation 2.9 is estimated with large finite values of n. Another way to describe entropy is to say that a recogniser will need H bits of information (if logs are to base two) in order to predict the next word in the sequence w_1^n. If the source is ergodic, then the value can be said to correspond to the difficulty which the recogniser has in predicting any next word for the whole of the underlying language. If we use the relation mentioned earlier between H and PP, we can conclude that PP can be calculated as follows

$$PP = P(w_1^n)^{-\frac{1}{n}} \tag{2.10}$$

This will be true for any language model, where a language model is defined as any process which can deliver an estimate for the value of $P(w_1^n)$.

Bahl, Jelinek and Mercer [6] demonstrate that there is a strong connection between word prediction error rate and perplexity, which provides a further argument for the use of perplexity in statistical language model evaluations.

2.7 Implementing a Corpus Formatter

There are two main parts to the implementation of any language model. First, a database needs to be constructed which contains the raw occurrence frequencies of all of those word segments which occur with a significant frequency. Second, an algorithm needs to be created which iterates through an input word sequence — the test set — calculating conditional probabilities for each word token by accessing the database to get frequency information.

According to O'Boyle's method, there are four processing stages involved in taking an unformatted corpus and constructing from it a segment frequency database. These are : formatting, segment generation, segment sorting and segment filtering.

2.7.1 Formatting

Work which involves the manipulation and analysis of large text files requires that either all of these files are in a standard format or else that a program is available which can produce corpora to certain desired specifications. Even if many corpora do conform to more or less standard requirements, there still are different types of corpora (*e.g.* tagged and untagged, orthographically edited or phonemic discourse; also, in text files, the issue of how to deal with punctuation currently allows for much variation – see, for example, figure 2.1).

A corpus formatter can be defined by what it expects to find in the original corpus (the raw corpus) and what it is expected to produce in the processed corpus.

Finite State transducers (FST's)

It was decided to use the object oriented nature of C++'s stream system to produce a formatter which transduced, one character at a time, from the raw to the formatted file. In order to make it as flexible as possible in the time available, a general-purpose recogniser and a reasonably general producer were written.

Example : VODIS

The VODIS corpus is a transcription of recorded telephone conversations made to a British Rail Enquiries office at Ipswich; each sentence starts with a number and may or may not contain newline characters. One transducer was designed to remove these numbers and newline characters. It also isolates punctuation symbols from words so that they can be dealt with separately.

The requirements of the formatted corpus are influenced by the uses of the finished language model. For example, with a language model which is to be used with an optical character recogniser, minimal formatting is indicated; in particular, punctuation positions are left unchanged. If the language model is destined for operation within a speech recognition system, then punctuation symbols must be considered carefully.

While the corpus can be seen as related to the utterances which generated it, generally corpora can come from any sources : they can be fully textual documents which have never been uttered. Also, any orthographic regularities which exist in the corpus — for example the punctuation symbols which attempt to capture prosodic features of language — must be regarded as part of the structure of the corpus. Removing them might lead to a prediction task which is markedly harder. If all letter-case information and all punctuation information was removed from the corpus, the resulting corpus would be a more difficult one to model. The finite state transducer can also remove word tags from a tagged corpus. A detagger was written for the LOB corpus, which was used in some experiments described in chapter 5.

2.8 Experiment — Effect of Corpus Format on Perplexity

Prosodic elements of a speech utterance provide humans with many cues to the structure — syntactic and semantic — of the speech event. Much of this is lost whenever language is codified in symbolic forms, including most computer readable corpora. There are, however, some conventions which writers of language use to try to capture, in an admittedly coarse-grained way, some of these para-linguistic features. Some important conventions are the full stop, the comma, the dash, the exclamation mark, the question mark, *etc.* A corpus was created from which all punctuation was removed and all ASCII letters were made lower case; we produced a second alternative version of the corpus, which retained punctuation, but converted all letters to the same case. These two alternatives, plus the standard corpus were split up into 10% test sets and 90% training sets.

The weighted average language model described by O'Boyle *et al.* [121] was chosen as

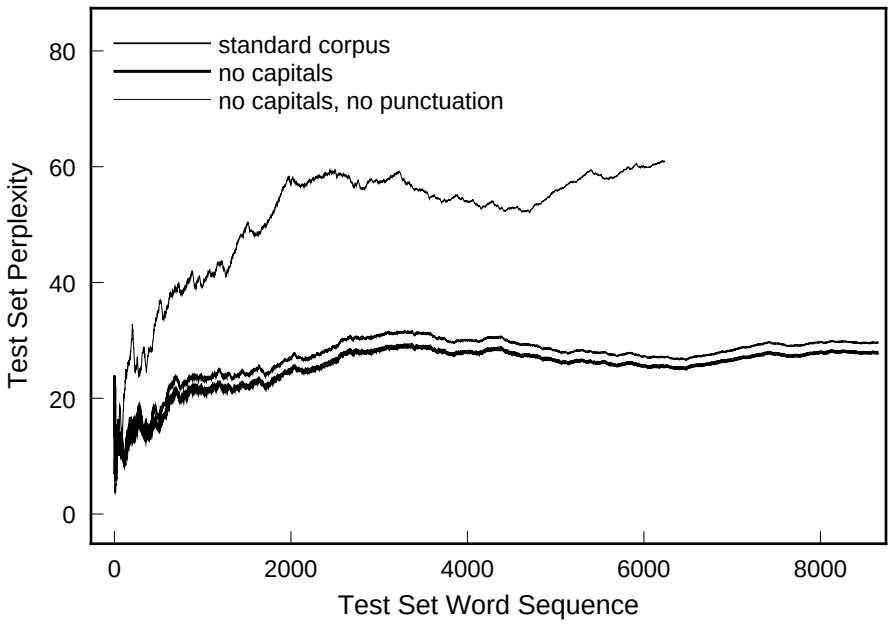


Figure 2.1: Comparison of the perplexity scores, using similar language model, test and training sets in all cases, of three differently formatted corpora.

the language model of preference for this experiment — it was implemented in C++ and was used as the probability component of a perplexity estimator, also implemented in C++. The weighted average model combines maximum likelihood probability estimates of unigram, bigram, trigram and generally any n-gram segments for which sufficient statistics exist.

Figure 2.1 shows the ongoing perplexity score as each word of the three test sets is input to the same weighted average language model. The most obvious feature of these graphs is that the corpus with no punctuation results in a much higher test set perplexity. This can partly be explained as follows. The first reason is relatively uninteresting : punctuation symbols account for a significant percentage of 'vocabulary' of the original corpus. Thus the full stop, for example, has a relatively high probability of occurrence in both the test and training sets. Any language model based on data from the training set will estimate $\hat{P}(\langle . \rangle)$ to be relatively high. This is mirrored in the test set, where we can assume that the frequency of occurrence of the full stop is just as high. Thus, for every full stop that is encountered in the test set, a relatively high probability value will be incorporated into the test set probability — and hence the test set perplexity. A second reason why punctuation might lead to a lower perplexity is possibly because we are dealing with a highly formalised domain — one where the telephone operator almost always starts the conversation off with ⟨`Hello`⟩ or ⟨`Good Morning`⟩ or ⟨`British Rail Enquiries`⟩; consequently punctuation — especially the full stop — might be a useful predictor of subsequent segments. This tends

to contrast with the effect of punctuation in so-called free discourse, where it often has the role of marking constituent boundaries (see [157, 14, 45, 159] and section 4.3.2 for examples); at these boundaries the predictive uncertainty tends to rise. The former reason is the more likely to explanation, though the idea that the more constrained the domain, the lower the uncertainty is still a valid one. In a culture, one can think of social situations which are highly constrained as being easier to predict; they could be considered to be contexts which constrain the set of possible actions available to a human. This set is obviously normative and not at all prescriptive — a person might very well act unusually in a particular social context in just the same way that a person might utter an ill-formed sentence, or a meaningless one, or utter a well-formed sentence which includes a particularly rare word. But it is still the case that, over significant numbers of observations, normative patterns emerge.

The second piece of information which this figure tells us is that the system performs only slightly better whenever all of the letters are in the same case. This could be explained by the fact that, whereas previously the words ⟨`That`⟩ and ⟨`that`⟩ were considered different, now they count as one word – so the overall vocabulary is smaller and an entropy measure should be correspondingly less. Interestingly, Church *et al.* [30] note that, with certain polysemous words, case can be useful in disambiguating the proper sense; for example, the word ⟨`bank`⟩ has at least two distinct senses — river bank and money bank. The capitalised ⟨`Bank`⟩ is much more often associated with the money bank sense of the word; so retaining capitals also retains potentially useful linguistic information. Similar phenomena are reported in section 3.5.1.

Finally, figure 2.1 gives an indication of the variability in perplexity through the test set — this supports the use of significantly sized test sets if the resulting perplexity scores are to have any relevance on the language from which the corpora are drawn.

2.8.1 Implementation Details

The only data structures needed to define an FST are symbol streams and node structures. C++ already supplies a rich stream manipulation system, but the FST required a slightly lower level of interaction with streams. A class of object called `Node` was designed which contained all of the information and data manipulation necessary to implement an FST. Any node, N can be described as a set of arc-definitions of the following type

$$N = \langle \alpha \rightarrow \beta, N' \rangle$$

where α is a symbol recognised in the consumer, β a symbol output to a producer and N' is the location of the node where control passes after the first stream has consumed α and the

second has produced β.

An FST for the VODIS corpus

The following FST definition produces an output corpus in which :

- words are separated by spaces
- there are no newline characters
- there are no sentence numbers
- punctuation is split away from any word to which it may have been attached (*e.g.* ⟨`was true because, he said`⟩ becomes ⟨`was true because , he said`⟩).

In the following, the ⊔ symbol represents a blank space, and c, n, p and e represent characters, numbers, punctuation symbols and end of line markers respectively. The # character here represents null-output.

$$N_1 = \langle n \rightarrow \#, N_1 \rangle$$

$$N_1 = \langle p \rightarrow \sqcup + p, N_2 \rangle$$

$$N_1 = \langle c \rightarrow c, N_2 \rangle$$

$$N_2 = \langle p \rightarrow \sqcup + p, N_2 \rangle$$

$$N_2 = \langle c \rightarrow c, N_2 \rangle$$

$$N_2 = \langle n \rightarrow n, N_2 \rangle$$

$$N_2 = \langle e \rightarrow \sqcup, N_3 \rangle$$

$$N_3 = \langle p \rightarrow \sqcup + p, N_2 \rangle$$

$$N_3 = \langle c \rightarrow c, N_2 \rangle$$

$$N_3 = \langle n \rightarrow \#, N_1 \rangle$$

$$N_3 = \langle e \rightarrow \#, N_3 \rangle$$

$$N_4 = \langle \rangle$$

These four nodes constitute the formatter which processes the VODIS corpus. The End node (N_4) exists, but does not need arc details, because the program always and only sends control to the End node whenever the end of the input/consumed corpus is reached.

This transducer takes approximately 30 seconds to process the formatted VODIS text, which is 86,588 words long.

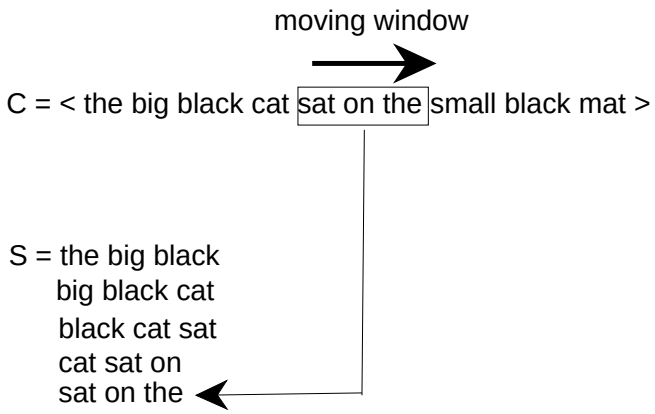


Figure 2.2: The corpus, C, has a three-word window passed over it. This window marks off all of the segments which will be operated upon by later stages of processing.

2.9 Processing a Formatted Corpus

The following four stages transform a formatted corpus into a functioning n-gram segment database.

2.9.1 Segment Generation

A window of fixed width is moved along the formatted corpus, a word at a time, and at each point, the encompassed segment of words is added to the list of segments generated by this process (see figure 2.2). The width of the window effectively limits the maximum length of segments to be considered; this width can be set to as large a value as is desired. The end product of this stage is a list of segments of equal length, plus a small number of segments from the end of the corpus.

2.9.2 Segment Sorting

The set of segments, which is stored as an ASCII file, with one segment per line, is sorted by any fast sort algorithm which can deal with large files. The UNIX sort facility provided the basis for the current sort implementation. The files are large and so need to be split, sorted separately and then merge-sorted.

2.9.3 Segment Filtering

Once the segments are sorted, we are in a position to count their frequency of occurrence and to implement a filter which picks only those segments which meet whatever requirements

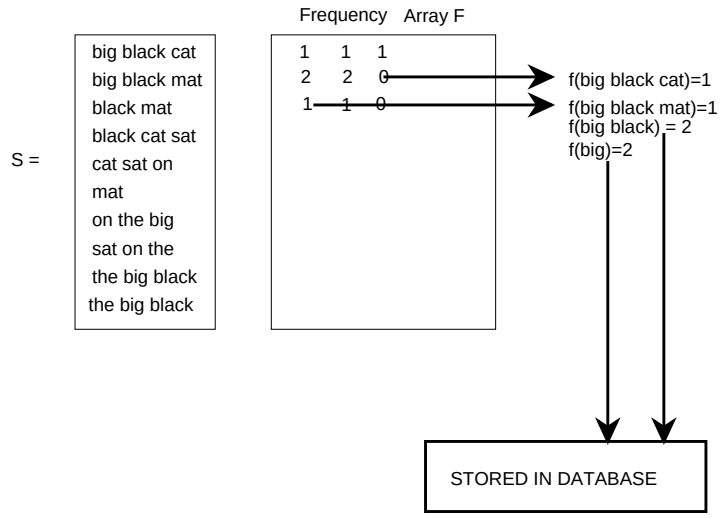


Figure 2.3: A sorted segment list allows for easy calculation of the occurrence of all of the sub-segments contained in the original full segment list. Each frequency array element in F represents the frequency of occurrence of that sub-segment which is made up of all of the words in that row, up to the column position where the frequency score is stored.

we care to set (see figure 2.3). This combination of sorting and selecting segments whose frequencies are greater than a set threshold is due to O'Boyle [119].

2.10 Summary

Statistical language modelling estimates segment probabilities based upon frequencies provided by corpora. The sparseness of segment frequency information has led researchers to exploit those segments whose frequency information is more reliable. Towards this end, it has been shown that perplexity is a useful measure of language model performance.

As an illustration of the power of the weighted average language model, nine versions of the oronym introduced at the start of the chapter were given as input to a weighted average language model, trained on a medium-sized corpus. Figure 2.4 shows the resulting probability estimates. Of the hypothesised sentences, the weighted average model ranks the original utterance as its second choice. The word ⟨`buoy`⟩ did not occur in the corpus, and so was undefined, leading to a zero probability estimate for the sentences which contain that word. This is another illustration of the sparse data problem. When we look closely at the sentence which was ranked better than the preferred sentence, we see that the only difference is in the two segments ⟨`boys eat`⟩ and ⟨`boy seat`⟩, neither of which occurs in the training corpus. Therefore the corresponding conditional probability estimates are made using unigram

sentence	probability $\times 10^{-20}$
the boy seat the sandwiches	3418.88
the boys eat the sandwiches	1787
the boy seat this and which is	435.392
the boys eat this and which is	231.73
the buoys eat the sandwiches	194.805
the buoys eat this and which is	25.3
the boys eat the sand which is	13.7556
the buoys eat the sand which is	1.499
the buoy seat this and which is	0

Figure 2.4: Nine versions of a phonemically identical oronym, ordered by probability. The weighted average language model ranks the preferred sentence above all but one of the less favoured options

frequency information only. While the unigrams ⟨`eat`⟩ and ⟨`seat`⟩ have approximately the same frequencies — 33 and 29 respectively — the unigrams ⟨`boy`⟩ and ⟨`boys`⟩ occur with significantly different frequencies 153 and 88 respectively. In other words, the weighted average language model based on word frequencies only falls prey to the sparse data problem with respect to the two zero-frequency bigrams and its subsequent reliance on unigram statistics distorts the overall sentence probability estimation. We shall return to this example in a later chapter.

The possibility of using other information about word segments — in particular, the frequency of occurrence of the various class segments of which they might be examples — could lead to language model combinations which are measurably better. The first stage of this investigation is undertaken in the next chapter.

Chapter 3

Class Based Language Models

3.1 Overview

We will now introduce a framework within which both class segments and word segments can be commonly represented. This framework also describes the limits of possible research work using class based language models. The discussion will then move on to a specific research area and a more detailed description of it. Four types of language model will be discussed in terms of the four database types from which they draw frequency information.

Next, two word classification methodologies will be briefly introduced, and a description given of the classification system used in the experiments described in this chapter. The first few experiments involve detailed analyses of the relationship between the frequency of a segment and the number of classes which follow that segment. The next experiment estimates some test set perplexity values for two simple class-based language models. Then, the idea of a binary classification of words shall be introduced through an experiment which calculates a series of related perplexity values. This idea will be extended in the penultimate section, where the *structural tag* is introduced. Finally, we shall draw some conclusions from the our work with the simple classification system.

3.2 Theory

Statistical language models can be described as conditional probability estimators. A set of elements — words or classes of words — are the events to be predicted, based upon the previous occurrence of a string of other elements — the context.

It is an implicit principle of statistical language modelling that the broader the context, the more successful the model. The sparseness of the data, however, means that there is a

huge variety of unique contexts, and uniqueness is antipathetic to statistical approaches.

If context was described fully it would include extra-linguistic elements, like discourse setting, domain details and, ultimately details about the person who is uttering the sequence of words, which itself relates to a general cultural context [106, 113, 158]. Clearly, at each point, the full context, if it could be completely specified, would be a unique structure and, as such, would provide no useful information for the task of predicting the next utterance. In order for contexts to be useful at all, there needs to be another way of looking at them which doesn't describe them as a potentially infinite sequence of structures in a one-to-one mapping with the next word. This is achieved by considering classes of context which are said to be equivalent, with respect to a particular task — for example speech recognition.

The n-gram language model represents one often used and powerful approach to generating equivalent contexts — the 'recent word context' equivalence class. It is commonly used partly because it involves an easy method of generating useful equivalent contexts whilst at the same time improving predictive power [95].

The use of equivalence classes allows a more general theoretical description of contexts. For example, the trigram models, which provided such a spur to research [32], can now be re-described in terms of that set of equivalence classes which unites all contexts which share the same last and second last word. The idea can be extended easily for n-gram models [120, 85, 84, 86].

3.2.1 Tree-Based Contexts

Bahl *et al.* [5] suggest an even more interesting theoretical description within which the idea of contexts can be situated. They argue that the conflation of contexts into equivalence classes which only deal with the previous n occurrences — *i.e.* n-gram language models — are useful but unnecessarily restrictive. For them, the design of a context structure should allow it to contain information about words at an arbitrary distance from the current prediction position, provided that this information is statistically reliable.

Recent-history equivalence classes based upon word information only allow for questions of the type : 'was the last word *the*?' or 'was the second-last word *cat*?' With their new context structure Bahl *et al.* have re-described these questions in terms of a binary decision tree associated with a set of heavily constrained question types — that is, a set of questions which only concern the last n elements of the context and which are only interested in identifying them as being identical to various vocabulary items. Bahl *et. al.* go on to describe the space of possible binary decision trees and to suggest methods of discovering some of the more

useful ones.

3.2.2 Contexts which Provide Information about recent history Word-Classes

The space of possible contexts is enormous; that region of the space which deals with recent-histories of words has been well investigated. The current research focuses its attention on another, less well investigated region — that which retains an interest in very recent context history, but which substitutes questions of the type 'was the last word a verb?' for questions like 'was the last word *runs*?' More generally, it investigates the distribution, in recent history, of classes of word. This includes, but is not limited to classification systems which theoretical linguists might recognise [18]. Jelinek [85] outlines the use of grammatical classifications in language models; Derouault *et al.* [42] combine grammatical and Markovian derived contexts and Resnik [135] investigates some of the advantages of class-based statistical language processing.

3.2.3 Intra- and Inter-Model Investigations

In any investigation of a portion of the space of possible language models, two central domains of interest can be distinguished. First, specific models can be researched with respect to how well they perform by themselves. In this case, the properties of the pure model can be described and compared with other models, using a universal metric, the test set perplexity. These issues can be referred to as intra-model investigations — experiments in this chapter provide some examples.

The second approach involves discovering how two or more models can be joined together to produce a further model. The main issues concern the degree of weighting of the different models — most of the experiments of chapter 7 involve this type of investigation. The distinction is made mainly for expository purposes, since often many pure models involve probability estimations which can be regarded as applications of simplified sub-models. Thus, for example, in the relatively pure domain of trigram language modelling, use is made of a model which specifically concerns itself with the distribution of rare segments (hapax legomena), and words which do not appear in the training set, whilst a simpler sub-model is applied for segments whose frequency statistics are more healthy [31, 58, 67, 73, 84, 86, 89, 114].

Generally, pure models exploit particular insights into the distribution of natural language utterances; these may sometimes come from theoretical discourses. If they are useful, they tend to model some aspect successfully. No theoretical insights exist today which lead to

models which capture all of the various aspects of language. As a consequence, in order to improve the performance of particular systems, it seems natural for researchers to incorporate disparate insights into their systems, which involves finding useful ways of relating the various models that have already proved to be at least partially useful.

3.3 Databases Containing Class information

That region of the space of possible contexts which this work aims to explore can be further sub-divided into four main regions, corresponding to four kinds of language model, which use four related but distinct types of information.

The first is the familiar n-gram word context, using a database which contains frequencies of word segments — a WW-database. The second makes use of a database which contains class frequencies — a CC-database. The third and fourth are hybrid databases, containing frequencies of 'all-word plus a class' segments, or 'all-class plus a word' segments — WC and CW databases respectively. For example, for the segment ⟨`the cat sat`⟩, the WW-database could contain an entry indicating how many times ⟨`the cat sat`⟩ occurred; the CC-database could tell us how often ⟨`det nn1 vvd`⟩ occurred; the WC-database could tell us how often ⟨`the cat vvd`⟩ occurred; and finally, the CW-database could tell us how often ⟨`det nn1 sat`⟩ occurred.[1]

Several general points can be made about the four database types described above. First, it should be remembered that one of the main reasons for looking beyond all-word language models was a realisation of the unwanted effects of sparse data — a situation where the vast majority of the contexts are unique. It should then become obvious that the WW-database and the WC-database, whilst providing fine-grained information, succumb to the problems associated with sparse data in similar ways. The WC-database is slightly better than the WW-database, but only just. The other two databases — CC and CW — offer the best hope of usefully attacking this problem. It should be noted that, even when a very coarse-grained classification system is picked — that is, one with few classes — the WC-database will *still* succumb to the sparse data problem, since each context is defined largely by words, and only in one case in n by a class; whereas a CC-database is defined by n classes, and hence any reduction in the number of classes will see a substantial reduction in the number of different contexts in the database, and a corresponding increase in their statistical significance. Several of the experiments of this chapter offer support for this hypothesis.

[1] where `det` is a determiner class, `nn1` is a singular noun class and `vvd` is the past tense of a lexical verb. Appendix A contains a full listing of this classification system.

The function of a prediction system is to predict *words*. Thus, the WW-database and the CW-database will be more directly applicable to the prediction process than the WC- or CC-databases, both of which will need an extra probability estimation — one which describes how likely a word is, given a particular class. When this extra probability estimation is a simple context independent one, the overall perplexity should always rise. That is, the performance of pure class-based language models will always be poorer than, or at best no better than word-based systems. The following section examines this point more closely.

3.3.1 Predicting Words from Classes

Given that a class has been predicted in a segment stream, the WC- and CC-databases will need to be able to estimate how likely a particular word is. Often, it is this sub-model which tends to increase the overall perplexity, since most estimates use only very constrained contexts. For example, the least informationally useful way to estimate how likely a word is, given a class, is

$$\hat{P}(w|C) = \begin{cases} \frac{1}{|C|} & \text{if } w \in C \\ 0 & \text{otherwise} \end{cases}$$

Where $|C|$ is the type size of C. A more sophisticated estimate sets the conditional probability estimate as follows

$$\hat{P}(w|C) = \begin{cases} \frac{f(w)}{f(C)} & \text{if } w \in C \\ 0 & \text{otherwise} \end{cases}$$

Both of these models estimate the word using only the fact that a class C has occurred. Also, both assume that a word can only belong to one class. The second model uses *a priori* global frequency estimates for the word and the class. Later in this chapter, we implement models which use these two word-prediction components. Other models could be investigated; for example, a model could be built which estimated the likelihood of a word, given not only the class, but also some of the recent classes. Another model could be investigated, described in Kuhn *et al.* [95]. The model is based upon the insight that the *a priori* likelihood of a word, given a class, is too generalised a measure of how likely any word is. If the words ⟨`band`⟩ and ⟨`man`⟩, for example, are predicted, based upon *a priori* frequencies, as p_1 and p_2 respectively, and if, as one might expect in an English natural language corpus, $p_2 > p_1$, then the standard model would always prefer ⟨`man`⟩. But if the previous recent utterances have included the word ⟨`band`⟩ many times, and ⟨`man`⟩ not at all, then one would like to use a model which would increase the probability of ⟨`band`⟩ being chosen. This is precisely what Kuhn *et al.*'s *Cache-Based* system does. It keeps a record of the previous m word utterances belonging to a class,

for each class. It can then use this to provide a local estimate of which word is most likely, given any of these classes. This model can then be integrated with the background *a priori* model; the resulting hybrid $\hat{P}(w|C)$ estimator leads to a dramatic three-fold improvement in perplexity.

3.3.2 WW Database

The first database we shall describe is the standard segment-count database which, by various language model implementations [84, 19, 5, 58, 31, 85, 89, 114, 86]) directly estimates $\hat{P}(w_1^n|w_1^{n-1})$. One such language model implementation is described in [122]. Here

$$\hat{P}(w_1^n|w_1^{n-1}) = \sum_{i=1}^{n} \lambda_i \times \frac{c(w_i^n)}{c(w_i^{n-1})} \tag{3.1}$$

where

$$\sum_{i=1}^{n} \lambda_i = 1$$

and $c(w)$ represents the count of the word segment w. The results of some experiments involving the WW-database are given in chapter 7.

3.3.3 WC Database

Given a word context, the probability that the next word will be w can be estimated from the likelihood that category C follows the word context multiplied by the likelihood that $w \in C$. That is,

$$P(w_1^n|w_1^{n-1}) = \sum_{C} P(w_n^n|C_n^n) \cdot P(w_1^{n-1}C_n^n|w_1^{n-1}) \tag{3.2}$$

Models of this type share some similarities with co-occurrence smoothing methods [71], where a context is considered relevant to the prediction of a particular word w insofar as it occurs with some other word w' and w' has a similar distribution to w. The measurement of distribution similarity is not symmetrical. The greater the similarity, the greater weight that context is given in the conditional probability estimation. In the simple case where words have one class, the summation of equation 3.2 becomes trivial to calculate.

3.3.4 CW Database

Jelinek [84] suggests that word categories could be used as context in the determination of probability estimation.

This case is similar to the word-given-word model. The basic conditional word probability is estimated as follows

$$P(w_1^n|w_1^{n-1}) = P(C_1^{n-1}w_n^n|C_1^{n-1}) \tag{3.3}$$

In order to implement this system, two databases will need to be constructed which return counts for all-class segments and also for class-word segments. The information delivered by this system will be different from that returned by the previous system because it is less specific — more contexts will now be considered similar — but more statistically significant : the contexts will occur with greater frequency (assuming that N_c, the number of classes, is substantially less than V, the word vocabulary size). This type of class-based language model has not often been implemented by statistical language modellers.

3.3.5 CC Database

The CC system is described by the following equation :

$$P(w_1^n|w_1^{n-1}) = P(C_1^n|C_1^{n-1}) \cdot P(w_n^n|C_n^n) \tag{3.4}$$

Most of the experiments in chapter 7 are designed to use these databases.

The last two types of database try to minimise one of the two main limitations of second order Markov models as language models — the problem that many contexts are unnecessarily fragmented because they happen to consist of words which, while different, are not syntactically different, or more generally, not linguistically different. The other problem involves the rather short memory which second order Markov models offer. Investigative work has been carried out on selectively extending the memory of finitary language models by O'Boyle [121]. This thesis contains work which attempts to limit the other shortcoming to word-based language models — unnecessary fragmentation.

3.4 Classification Methodologies

In order to describe fully language models which make use of word classes, we must give consideration to different types of classification. Initially, the most linguistically appealing type of classification is one which maps words to the familiar syntactic categories of the theoretical linguist. Sampson [140] argues that these classification systems might not be the best with respect to the task of predicting which word comes next; for him, they tend to represent the preoccupations of professional linguists rather than the needs of corpus researchers. He goes on to suggest that these researchers should use those classification systems, regardless of how linguistically or psychologically plausible they are, which are the most useful to

corpus linguists. This advice is supported by experimental evidence, in Jelinek *et al.* [86], which reports that traditional part-of-speech classifications contained grossly unequal class frequency distributions; the authors conclude that classification systems based upon theoretical linguistic considerations produced information of possibly doubtful value for statistical language modellers. They describe a self-organising classification system (see [85] also) which, when interpolated with a trigram model, leads to a significant improvement in perplexity. Sampson, while in agreement with the idea of a classification system which is constructed from pragmatic as well as theoretical linguistic considerations, tends to favour a top-down approach — for example, the Susanne classification [140].

The experiences of the research community suggest that investigating ways of automatically deriving word classifications from corpora could be both fruitful in terms of improving language model performance and also instructive in terms of developing an understanding of linguistic behaviour, and how this might relate to the more formal grammatical descriptions of theoretical linguists. Not all computational linguists agree — Church *et al.* [30] prefer to emphasis semi-automatic investigations.

Using word classes does not guarantee an improvement in language model perplexity, even if it is assumed that all of the parameters of the model were set optimally. For example, if a classification system is used which includes a separate class for every word, then no improvement would be possible. Also, if a classification system had just one class, containing the entire vocabulary, no improvement would be possible. Even if the system fell between these two extremes, there is still a large set of classification systems which would not lead to an improvement in the performance of the predicting system. For example, if the words are randomly mapped to classes, then one would not expect any improvement in the performance — except in the cases where the mapping has, by chance, discovered some feature of the language which the prediction system can exploit in a systematic way.

There must be an important trade-off to be considered when classification systems are compared; as one of their main applications is in diminishing the extent of the sparse data problem, it seems initially plausible to pick a classification system which associates many words to few classes — that way the statistics which can be generated using that classification system are more likely to be significant. The main drawback with such a classification system however, is that the model becomes more confident about what the next class will be, but correspondingly less certain about particular words of that class.

3.4.1 Implementing a Simple Classification

It was decided that, for the preliminary experiments described in this chapter, a classification system would be applied to the word types of the VODIS corpus, in order to explore the distribution of classes. In this classification system, every word is assigned a single class; the system is based on the CLAWS system, described in [61] and has been extended to include punctuation symbols and place names. A full listing of the classification system can be found in appendix A, along with short explanations.

The task of deciding to which class each word in the vocabulary belongs was performed manually; a data structure was designed which allows a list of classes to be associated with each word. Also, a tool was created which allowed details for each word to be edited. The list of classes for every word contained only a single entry, namely that class which was most appropriate for that word. There are obviously many problems with such an approach to word classification, but the system was considered suitable enough to use in the preliminary experiments described below.

3.5 Experiments

The experiments described in this section are based on a WC-database. They aim to illustrate the nature of the sparse data problem through practical example and to introduce the methods by which classes might be used to improve language models. Also, since much work has already been done on WW-databases and since chapter 7 of this thesis contains work which uses a CC-database, the present experiments have been designed to use the WC-database so that we will have implemented three of the four class database types introduced earlier.

3.5.1 Estimating the Class of the Next Word

The first experiment involves creating a database full of word segments, but with the last word in each segment replaced by the single class associated with that word. Thus, given the segment

```
British Rail
```

an algorithm that can access the new database can return with a list of possible next categories, together with their frequencies :

(aj0,1) (nn1,1) (nn2,80) (pnp,1) (prp,1) (pun,227) (vbd,1)

This produces statistics which could be used to augment the raw language model, which already finds

```
British Rail ,
```

and

```
British Rail .
```

to be the two most likely segments — where both ⟨,⟩ and ⟨.⟩ are classified as 'pun' (punctuation). The remaining part of this section describes in detail results for the segment :

```
how much for a second class return
```

plus the following distortions, which might occur if the utterance is presented to a speech recogniser :

```
how much for a second class is return
how much for a second class as return
how much for the second class is return
how much for a second glasses return
```

This example was taken from work on a simulated speech recognition system described in O'Boyle [119]. The following figures display the relevant results. Figure 3.1 shows the *a priori* frequencies of occurrence of all of the categories. Next, figure 3.2 shows what categories are expected if the word ⟨how⟩ is already given. Note that, by keeping the informationally-rich distinction between upper and lower case letters, differences are detected in the estimation of which categories and words come next. Thus, for ⟨How⟩, there is a slightly greater likelihood that the following word might be a verb than for ⟨how⟩ — *e.g.* ⟨How does⟩, ⟨How do⟩, ⟨How can⟩, ⟨How may⟩. Also, prepositions are more likely to follow ⟨how⟩ rather than ⟨How⟩ — *e.g.* ⟨tell me how to⟩ as opposed to the unlikely : ⟨How to⟩. The high likelihood of following adverbs is explained by segments such as ⟨how many⟩ and ⟨how much⟩, common segments in a train enquiries corpus.

The next figure — figure 3.3 — also illustrates an argument for keeping the corpus case sensitive. The segment, ⟨How much⟩, when input to the category prediction system, expects the class ⟨vbz⟩ — *i.e.* the single word ⟨is⟩. This result is to be expected, and is mirrored in the relatively high score of ⟨vbz⟩ for the lower case segment ⟨how much⟩. However, the class ⟨at0⟩ — referring to the indefinite article — is not predicted at all with ⟨How much⟩, since sentences starting with the phrase ⟨How much a⟩ or ⟨How much an⟩ should occur very

{ }				
(aj0,2782)	(ajc,161)	(ajs,96)	(at0,3389)	(av0,3292)
(avc,37)	(avp,397)	(avq,1146)	(avs,33)	(cjc,1486)
(cjs,1062)	(cjt,1530)	(crd,4967)	(dge,193)	(dt0,843)
(dtq,2)	(ex0,81)	(gen,1235)	(itj,4475)	(nn,7)
(nn0,34)	(nn1,6372)	(nn2,1007)	(nnp,14)	(np1,29)
(np2,7)	(npp,2731)	(one,567)	(ord,149)	(pge,2)
(pni,124)	(pnp,6964)	(pnx,17)	(pro,538)	(prp,6513)
(pun,21523)	(unc,252)	(vbb,1035)	(vbd,161)	(vbg,5)
(vbn,36)	(vbz,919)	(vbd,441)	(vdd,51)	(vdg,16)
(vdn,7)	(vdz,127)	(vhb,559)	(vhg,8)	(vhn,23)
(vhz,30)	(vm0,1800)	(vvb,4630)	(vvd,549)	(vvg,951)
(vvn,87)	(vvz,433)			

Figure 3.1: *A priori* category frequencies.

how				
(aj0,17)	(av0,83)	(ctj,3)	(dt0,8)	(pnp,2)
(prp,7)	(pun,2)	(vdb,5)	(vm0,2)	
How				
(aj0,14)	(av0,32)	(dt0,8)	(pnp,1)	(prp,4)
(pun,1)	(vbb,1)	(vbz,1)	(vdb,2)	(vdz,2)
(vm0,2)	(vvb,1)			

Figure 3.2: Categories which come after initial-sentence ⟨`how`⟩ and mid-sentence ⟨`how`⟩.

How much					
(pun,3)	(vbd,1)	(vbz,20)	(vdd,1)	(vdz,1)	(vm0,1)
how much					
(at0,18)	(cjt,1)	(dt0,2)	(pnp,22)	(prp,4)	(pun,2)
(vbb,2)	(vbz,12)	(vdz,5)	(vm0,9)		

Figure 3.3: Classes following upper and lower case ⟨`how much`⟩

How much for	how much for	much for a
much for the	for a second class	for the second
a second glasses	a second class is	a second class as
the second class	second class is	second class as
second glasses	class as	

Figure 3.4: Segments which returned no category suggestions. Only the smallest segments are repeated here : since ⟨`How much for`⟩ cannot predict any categories, ⟨`How much for a`⟩ returns nothing either.

infrequently; with the segment ⟨`how much`⟩, however, ⟨`at0`⟩ is the second most predicted category, since sentences which contain the segment ⟨`how much a/an`⟩ are much more frequent : *e.g.* ⟨`Could you tell me how much a return to London costs?`⟩. Also, the class ⟨`pnp`⟩ isn't predicted by ⟨`How much`⟩ at all, but is the most likely class following ⟨`how much`⟩. This corresponds to mid-sentence segments like ⟨`how much it would cost`⟩.

In these experiments, case sensitivity and punctuation are retained, but if a final language modelling system was to be used with a speech recogniser, for example, then a differently formatted corpus would be used.

Figure 3.4 illustrates the single biggest weakness in the current next-class estimation system; this figure shows all of those segments for which there wasn't enough information in the database to predict any next classes. In other words, many of the trigram segments, and almost all of the larger ones are unrepresented in the database. This shouldn't be surprising, since the database is just a slight variation on the original frequency database, where the sparse data problem lead to attempted solutions — for example, using the Turing-Good formula, backing-off and weighted averages. The implication of this weakness is discussed in more detail later, as are ways of creating a system which might give more useful results.

Next, figure 3.5 shows the list of categories predicted to come after the definite and indefinite articles ⟨`the`⟩ and ⟨`a`⟩. These words do not initially appear to be good at restricting the number of following categories. Instead of limiting the number of possible next categories to a handful, they each allow for about a half of all possible categories. When compared to the two most contentful words of the current example, ⟨`class`⟩ and ⟨`second`⟩ (see figure 3.6), their performance is poor — ⟨`class`⟩ predicts only seven, and ⟨`second`⟩ only six following classes. Part of the reason for the sheer variety of categories following articles lies in the fact that these words are so frequent in the corpus : the more frequent a word, the more frequent its contexts. When the frequencies are looked at more carefully, the expected domination of

the				
(aj0,283)	(ajc,9)	(ajs,60)	(av0,10)	(avc,3)
(avq,1)	(avs,23)	(cjt,2)	(crd,227)	(dt0,44)
(itj,1)	(nn0,7)	(nn1,934)	(nn2,202)	(npp,30)
(one,43)	(ord,38)	(prp,8)	(pun,51)	(unc,3)
(vbz,1)	(vvb,44)	(vvd,2)	(vvg,6)	(vvn,1)
(vvz,6)	(zz0,2)			
a				
(aj0,142)	(ajc,14)	(at0,1)	(av0,27)	(avc,1)
(crd,18)	(dt0,8)	(nn0,1)	(nn1,835)	(nn2,3)
(np1,2)	(npp,6)	(one,1)	(ord,31)	(prp,15)
(pun,19)	(unc,3)	(vdb,1)	(vvb,61)	(vvd,3)
(vvg,2)				

Figure 3.5: The abundance of classes which can follow the definite and indefinite articles. Only the word ⟨`is`⟩, in this experiment, is more varied.

noun-like categories emerges — categories like ⟨`nn0`⟩, ⟨`nn1`⟩, ⟨`nn2`⟩, ⟨`np1`⟩, ⟨`npp`⟩, ⟨`one`⟩ and ⟨`ord`⟩. Adjectives and adverbs also account for a significant proportion of next classes — ⟨`aj0`⟩, ⟨`ajc`⟩, ⟨`ajs`⟩, ⟨`av0`⟩, ⟨`avc`⟩, ⟨`avq`⟩ and ⟨`avs`⟩. But, given the fact that these are only estimated frequency results, and that, for the rest of the classes — those which appear a small amount of times, and even those which do not occur at all in this database, their *language* frequencies should be higher, widening the set of next classes even more. Further analysis and discussion of this topic is provided later. Francis and Kucera [57] report that articles are good predictors of the following class : nouns typically follow articles 51% of the time, adjectives 20% of the time and the remaining 82 possible category types follow only 29% of the time.

Both figure 3.6 and figure 3.7 contain results which could be significant with respect to the main aim of the experiment. They show what the system expects after ⟨`class`⟩, ⟨`second class`⟩ and ⟨`a second class`⟩. In all three segments, the dominant category is that of the singular noun. Also, as the segments get longer — *i.e.* as the context gets wider — then the results of the prediction process become more specific : the number of next classes drops from seven to five to one. At the level of the trigram, the simple frequency system only predicts singular nouns. In fact, segments such as ⟨`Could you tell me how much a second class is`⟩ and ⟨`I think a second class is five pounds`⟩ re-emphasise the limitations imposed

'second'				
(aj0,2) (vvb,1)	(itj,2)	(nn1,17)	(one,1)	(pun,17)
'class'				
(aj0,1) (pun,11)	(nn1,19) (vbz,2)	(nn2,1)	(npp,1)	(prp,3)

Figure 3.6: Content words predicting a small number of next categories — although this may be explained in terms of sparse data, rather than any genuine property of the word.

second class				
(aj0,1)	(nn1,9)	(npp,1)	(pun,2)	(vbz,1)
a second class				
(nn1,7)				

Figure 3.7: Some segments predicting a noun-type word next.

by the sparse data problem. We want our predictor to be more flexible, which means that we want to find a way of counter-balancing this constant under-estimation of frequencies. Classes which are not predicted by a given segment ideally should have some non-zero estimation associated with them.

The other interesting point about the figures which predict what comes after ⟨`class`⟩ and ⟨`second class`⟩ is that, in both cases, the category ⟨`vbz`⟩ — corresponding to the word ⟨`is`⟩ — is predicted. This again should not be surprising, since ⟨`class is`⟩ and ⟨`second class is`⟩ are prefectly sensible bi- and trigrams. However, if the database was large enough, it is hoped that the segment ⟨`how much for a second class`⟩ should contain a low estimated frequency for the word ⟨`is`⟩ to come next. The main conclusion to be drawn from this is that the language model as it stands is still too heavily dependent upon the shorter segments, since they are much more frequent. When the word-based language model is run with the following segments, it can estimate the probability of every possible next word, and then sort this list of probabilities; in this case, we are only interested in the rank of the words ⟨`return`⟩, ⟨`is`⟩ and ⟨`as`⟩.

```
class return - 2nd
class is     - 7th
class as     - 124th
```

```
second class return - 3rd
second class is     - 5th
second class as     - 124th

a second class return - 3rd
a second class is     - 6th
a second class as     - 124th

for a second class return - 2nd
for a second class is     - 6th
for a second class as     - 124th

much for a second class return - 2nd
much for a second class is     - 6th
much for a second class as     - 124th
```

and so on. Any new information provided by 4-grams and higher is negligible, but trigrams and lower will always expect words like ⟨`is`⟩ to have a relatively high frequency. Thus, a language model such as this, using a corpus the size of VODIS, will not ever be able to provide information about a segment whose structure typically spans four or more words. In the present case, it is the segment ⟨`how much`⟩ which should decrease the likelihood of ⟨`is`⟩ being the next word, but it occurs too early in the example sequence for most n-gram language models to discover the dependency. The segment ⟨`how much`⟩ belongs to a different constituent than ⟨`a second class`⟩. The words ⟨`class`⟩, ⟨`second class`⟩ and ⟨`a second class`⟩ are all still a part of a noun phrase, so, knowing more about the noun phrase doesn't really help the system to predict what the next word will be, if it belongs to a different constituent, as ⟨`is`⟩ does.

The last two figures (figure 3.8 and figure 3.9) are of less importance to the general finding of this experiment, but are reproduced for completeness. Figure 3.8 displays similar results to figure 3.5, showing how words which are relatively low in semantic content and high in functional value do not initially seem to be particularly good at restricting the number of possible next categories.

is				
(aj0,23)	(ajc,7)	(at0,134)	(av0,33)	(avc,1)
(avq,10)	(cjc,1)	(cjs,2)	(cjt,49)	(crd,37)
(dge,4)	(dt0,5)	(itj,10)	(nn1,14)	(nn2,3)
(np1,1)	(npp,9)	(one,13)	(pni,2)	(pnp,171)
(prp,57)	(pun,111)	(unc,1)	(vbz,1)	(vdb,1)
(vvb,13)	(vvd,4)	(vvg,12)	(vvn,1)	
for				
(aj0,9)	(ajc,4)	(at0,140)	(av0,3)	(avq,3)
(cjt,21)	(crd,19)	(dge,26)	(dt0,11)	(nn0,1)
(nn1,14)	(nn2,12)	(npp,18)	(one,5)	(pni,4)
(pnp,64)	(pnx,2)	(prp,4)	(pun,30)	(vbz,1)
(vhg,1)	(vm0,1)	(vvb,1)	(vvg,3)	
much				
(aj0,1)	(ajc,5)	(at0,19)	(av0,42)	(cjc,1)
(cjs,7)	(cjt,1)	(dt0,2)	(itj,6)	(nn1,2)
(prp,22)	(pro,1)	(prp,15)	(pun,138)	(vbb,2)
(vbd,1)	(vbz,32)	(vdd,1)	(vdz,6)	(vm0,10)
(vvb,2)				

Figure 3.8: Classes following ⟨`is`⟩, ⟨`for`⟩ and ⟨`much`⟩.

for a				
(aj0, 4) (npp,1)	(av0, 2) (one,1)	(avc, 1) (ord,2)	(crd, 1) (pun,4)	(nn1,46)
for the				
(aj0,8) (npp,2)	(crd,3) (one,1)	(dt0,2) (ord,2)	(nn1,40) (vvb,2)	(nn2,8)
much for				
(at0,1)	(dge,7)	(pnp,4)		
for a second				
(pun,3)				
a second				
(itj,1)	(nn1,7)	(pun,14)	(vvb,1)	
the second				
(aj0,1)	(nn1,3)			
class is				
(pnp,1)	(pun,1)			

Figure 3.9: Next class prediction for the remaining relevant segments.

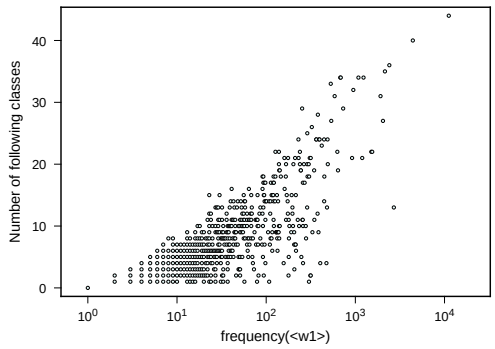


Figure 3.10: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a one word segment.

Conclusions

The limitation of the raw language model is its inability to contain information on medium and long distance linguistic features; it naturally performs well with short features like adjective-noun positions, preposition sequences, *etc.*, but medium distance features like some kinds of verb agreement — *e.g.* ⟨`the` *train* `now at platform one` *leaves* `at three o'clock`⟩ — are beyond its statistical reach, as are the more long-distance features of anaphor and discourse-related restrictions.

Given corpora of the size of VODIS, this sparse data problem is only partially dealt with by particular language models, for example, the weighted average one. Without increasing corpus size by several orders of magnitude, these types of agreement will always expose the limitations of all-word language models.

The current experiment is still essentially based on the frequencies of word-class hybrid-segments where only the last element of the segment is a class. As a result, all experiments of this type will fall prey to the problem of sparse data at around the 3- and 4-gram level.

If a CC-database was created (using the 57 tags of appendix A), this increases the size of statistically significant n-grams beyond 4. This extension, while loosing some of the fine-grained detail of actual-word prediction, instead allows a broader context to be investigated. As the number of classes decreases, the statistical significance of n-grams increases.

3.5.2 Segment Frequency and Number of Following Categories

An analysis of all of the segments in the VODIS corpus was carried out to discover the nature of the relationship between the frequency of particular word segments and the number of categories which the segments predict. The results are summarised in figures 3.10 to 3.15 for segments of various lengths. The first trend to notice across all six figures is that there is a

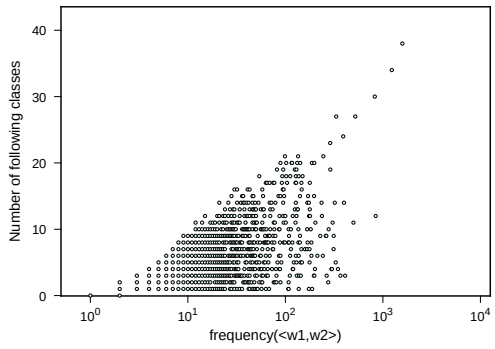


Figure 3.11: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a two word segment.

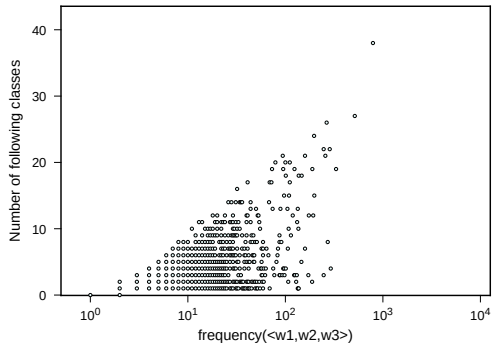


Figure 3.12: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a three word segment.

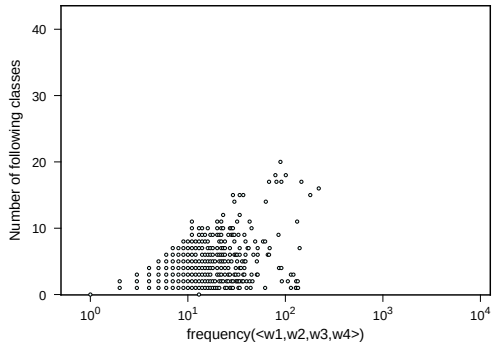


Figure 3.13: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a four word segment.

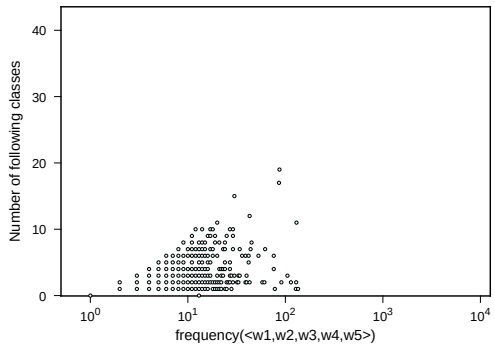


Figure 3.14: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a five word segment.

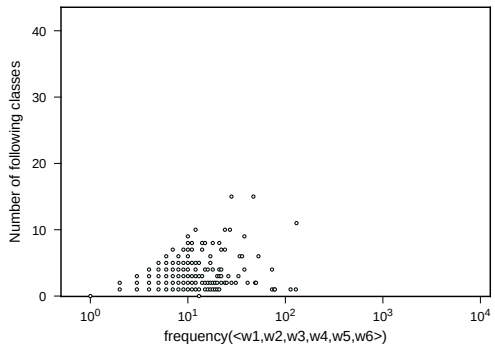


Figure 3.15: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a six word segment.

clear relationship between the frequency of segments and the number of following categories, a relationship which seems to hold, no matter how many words are in the segment. That is, six segments, of length $n = 1, 2, \ldots, 6$ which all occur in the database with the same frequency seem to have the same range of numbers of categories which can follow them.

The second trend concerns the nature of this spread of next category values at given frequencies. It is best to look at the four main frequency ranges of these figures. The range $10^0 - 10^1$ is the most consistent throughout the different segment sizes, and is easiest to explain : if any segment occurs with a frequency f in the range $1 < f < 10$, then the maximum number of categories which follow is fixed to the value of f.

The range $10^1 - 10^2$ shows a steady rise in the number of following categories and it is here that the spread of number of following categories is most plain : given that the relatively constant rise in the number of categories can be partially explained by the fact that the segments in question are occurring more and more frequently, and that the more a segment of given size occurs, the more categories are likely to follow it — *i.e.* given the positive and approximate log-linear gradient of the upper edge of the data points in the figures — the distribution of data points *below* this leading edge is wide. The final two ranges show a more sparse distribution. There are some difficulties in trying to interpret these graphs. It might be the case that the spread of following classes seen in the diagrams could be explained entirely by the frequencies of the new hybrid segments.

Another complicating factor which makes these diagrams difficult to interpret might be that the distribution of sequences might not differ in a statistically significant way from a classification system where words are assigned to classes randomly.

A third problem in plotting number of following classes against frequency is that no indication is given that, at any frequency and over the predicted range of numbers of following categories, there will be an uneven distribution of the numbers of frequency-class number points — all that this diagram records is the presence of at least one data-point at any particuar co-ordinate. Many data-points will be mapped onto the same co-ordinate in these two-dimensional figures.

A fourth problem with these figures is that this classification consists of categories whose membership varies. In other words, even for data-points which have the same predicted number of categories, this still gives no clear indication of the predicted word entropy. The point can be illustrated with a worst-case example. Two segments, each of which have the same length and frequency can predict one and twenty following categories, respectively. In the case of the segment which predicts one category, the membership of that category

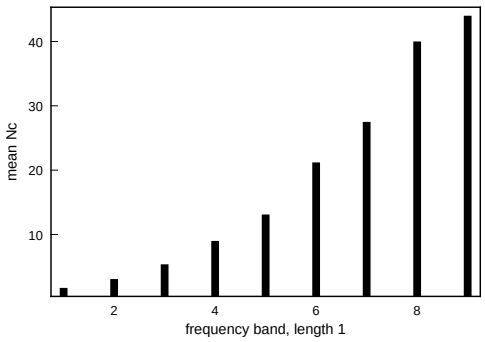


Figure 3.16: Mean number of following classes within frequency bands of width $\log \frac{1}{2}$, for one word segments.

accounts for only a quarter of the entire vocabulary. In the second case, the sum of the members of the twenty categories accounts for only one eighth of the vocabulary; yet from the figure alone, we might conclude that the first word-segment was better at reducing the entropy than the second.

Clearly, knowing how good word segments are at predicting word categories is not as informative as knowing how good word categories are at predicting particular words.

It was considered desirable to discretise the x-axis into frequency bands, and to calculate the average number of classes for each band — these results could then be displayed on a graph of $\overline{N_c}$ against frequency band, for all segments of a given length.

$\overline{N_c}$ against f-band graphs

The following piece of analysis was carried out with two band-widths, both of which are measured on a logarithmic scale. Figures 3.16 to 3.21 show the results with a band width of $\log \frac{1}{2}$; the experiment which used a band-with of $\log \frac{1}{10}$ resulted in finer-grained, but broadly similar figures, which are not repeated here. It was decided that the arithmetic mean would be the least distorting measure of the average N_c value. The mode, in most cases, would produce values of average N_c lower than the mean, since the spread of N_c values within any band is skewed towards $N_c = 1$. The median would have the opposite effect, due to the high degree of spread within these skewed frequency bands.

The method of calculating the means was as follows. All of the (N_c, f) data points belonging to a particular band are gathered together and the mean is the value of the expression

$$\frac{\sum N_c \times f}{\sum f}$$

summed over these points.

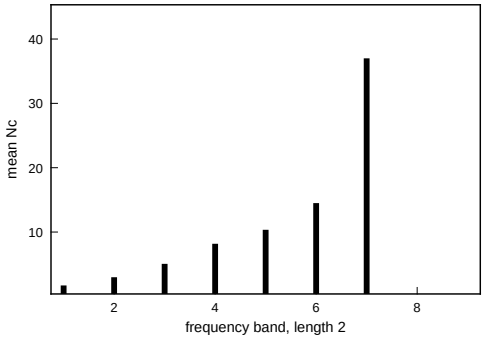


Figure 3.17: Mean number of following classes within frequency bands of width $\log \frac{1}{2}$, for two word segments.

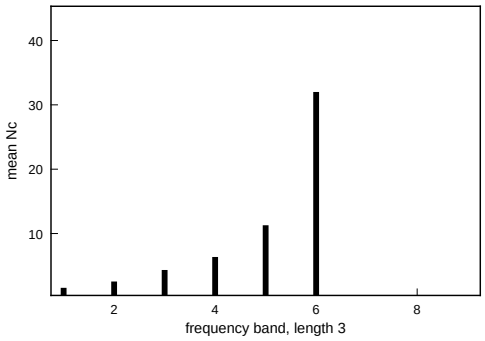


Figure 3.18: Mean number of following classes within frequency bands of width $\log \frac{1}{2}$, for three word segments.

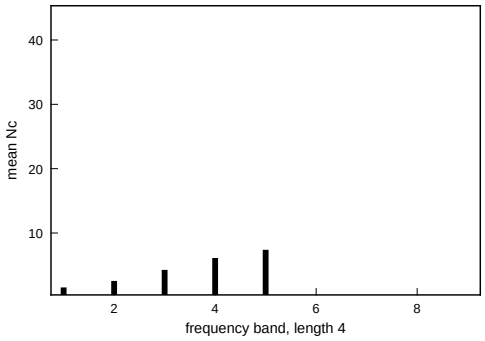


Figure 3.19: Mean number of following classes within frequency bands of width $\log \frac{1}{2}$, for four word segments.

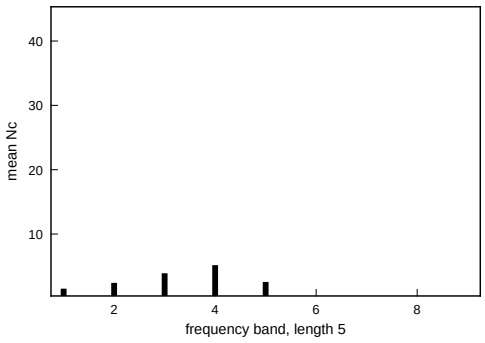


Figure 3.20: Mean number of following classes within frequency bands of width $\log \frac{1}{2}$, for five word segments.

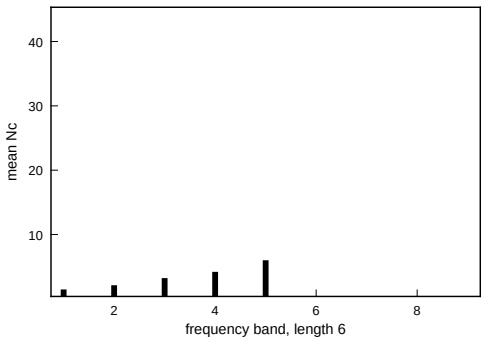


Figure 3.21: Mean number of following classes within frequency bands of width $\log \frac{1}{2}$, for six word segments.

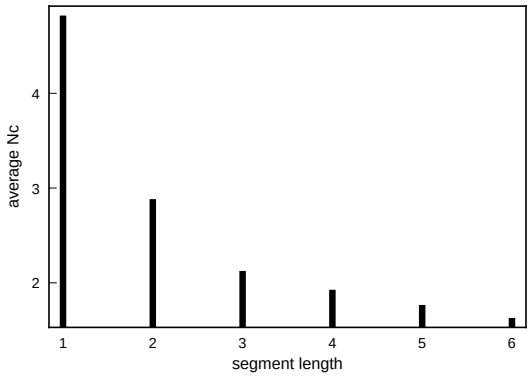


Figure 3.22: Overall mean number of following classes against segment length.

Mean ($\overline{N_c}$) against Segment Length

The values of $\overline{N_c}$ increase across the frequency bands, for each of the segment history lengths, but the significance of the separate $\overline{N_c}$ values depends upon the number of data points in the bands.

Figure 3.22 shows how the final estimate of the number of following classes varies against segment length; the values decrease with segment length. However, the conclusion cannot be drawn that, as the word-segment history increases, the degree of uncertainty decreases with respect to which classes follow. This is because we already know that as segment length increases, the database contains fewer examples of segments of this length, and that, of those segments that are present, they will occur with lower frequencies; if a segment of length six only occurs a relatively small number of times, then the final $\overline{N_c}$ value will always be small.

$\overline{N_c}$ against Segment Length across the same Frequency Band

In order to avoid comparing high frequency-based values with low frequency ones, a series of graphs can be drawn which show how $\overline{N_c}$ varies with segment length for values of $\overline{N_c}$ estimated within particular frequency bands. Figures 3.23 to 3.29 contain these graphs for bands of width $\log \frac{1}{2}$. The overall trend is for $\overline{N_c}$ to decrease as segment length increases. Bands 1 – 4 are based on statistically more significant data than bands 5 – 7, and in the first four bands, there is a monotonic decrease as segment length increases. There are too many bands to include here when a width of $\log \frac{1}{10}$ is used, but these graphs have been plotted, and similar effects are observed.

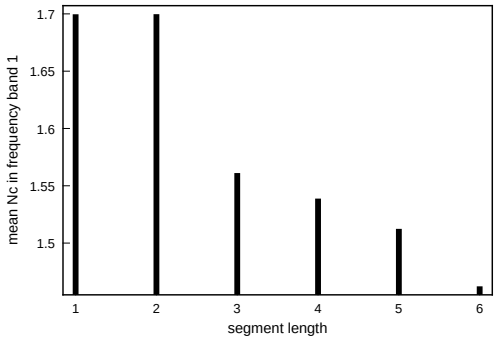


Figure 3.23: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 1.

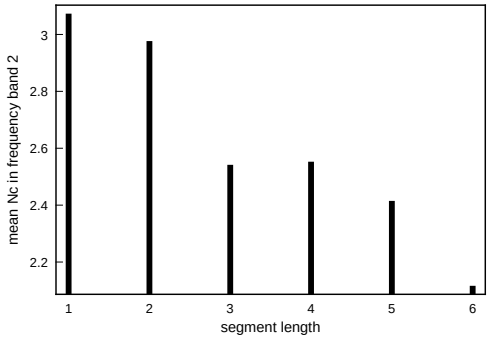


Figure 3.24: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 2.

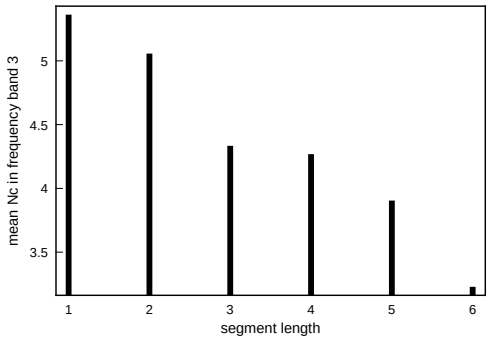


Figure 3.25: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 3.

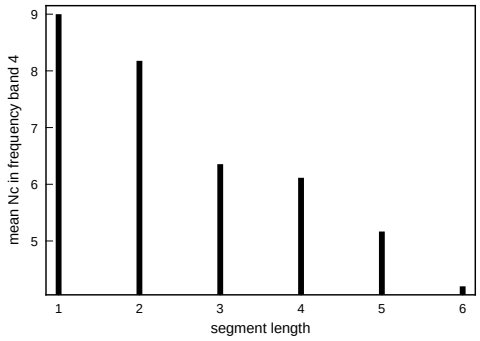


Figure 3.26: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 4.

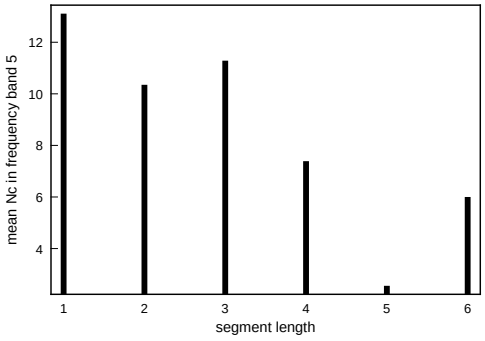


Figure 3.27: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 5.

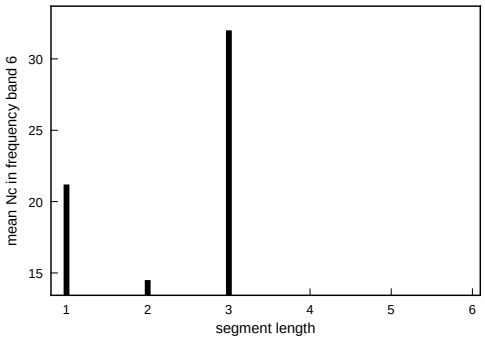


Figure 3.28: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 6.

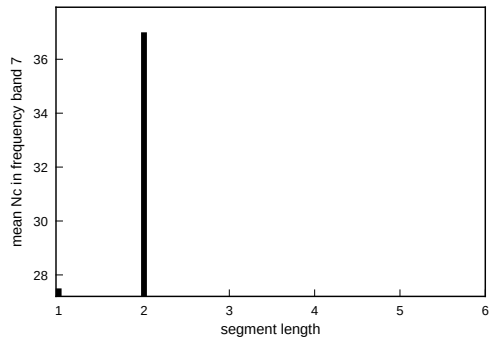


Figure 3.29: Graph of mean $\overline{N_c}$ against segment length, for segments whose frequencies fall within band 7.

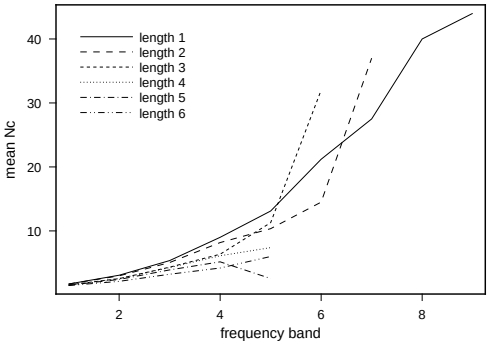


Figure 3.30: Graph of mean $\overline{N_c}$ against frequency band, for all six segment lengths.

Conclusion

The aim of this analysis was to show that, as the length of the word segment in the context — *i.e.* the part which would do the predicting of which classes follow it if it were to be used in a prediction system — increases, the number of following classes, on average, decreases. This has been demonstrated.

The graph in figure 3.30 shows a series of six lines, one for each length, which are made up by joining together the points which correspond to the $\overline{N_c}$-f-band number points. The final graph, shown in figure 3.31 is an example, for segment length 1, of the combination of the original $\overline{N_c}$ against f data set plus a line which is derived from points which correspond to the maximum boundary frequency for the various bands, as x-co-ordinate, and the $\overline{N_c}$ value for that band as y-co-ordinate. This analysis confirms one of the main principles of statistical language modelling — namely that the broader and more statistically significant a context is, the better a given system based on this context will perform.

The most useful insight which can be taken from this piece of analysis is an awareness of the sparse data problem. Whenever a context consists of words only, this problem has hardly

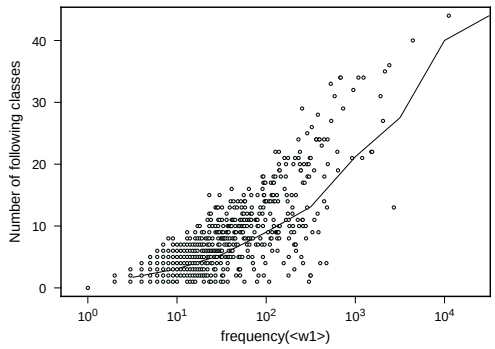


Figure 3.31: Graph of mean $\overline{N_c}$ against frequency, for segment length one, superimposed upon the original N_c-f graph.

been alleviated, since it is always the context which is doing most of the work. When the context is class-based we should make advances towards alleviating this problem.

3.5.3 Decreasing N_c

The test set perplexity is a useful indicator of how good the language model is. The model in this experiment has as its core the estimation of the conditional probability

$$\hat{P}(w_1^n|w_1^{n-1})$$

which it equates to

$$\hat{P}(w_1^{n-1}C|w_1^{n-1}) \times \hat{P}(w^n|C)$$

There are two distinct parts to this model, and any weakness in one will lead to a higher overall perplexity. If we concentrate on the second half of the model — the weak half, which is estimated as the maximum likelihood probability

$$\frac{f(w)}{f(C)}$$

we can see that, as the classification scheme gets broader, this estimate will produce smaller and smaller probabilities. If p represents the perplexity recorded using the old classification system of 57 classes, then we can see the effect as N_c is increased or decreased. As N_c tends to V, then the second half of the model will tend to 1, meaning that the first half is doing all of the work – at the limit, the first half will be acting as a simple all-word n-gram, and should give a perplexity value of less than p. As N_c tends to 1, the first half of the model tends to one, but the weak, *a priori* second model is left doing all of the work. This reduces, in the limit, to a context-free unigram language model, and its perplexity score should be higher than p. This corresponds to a word-emitting source whose non-equiprobable words are emitted independently.

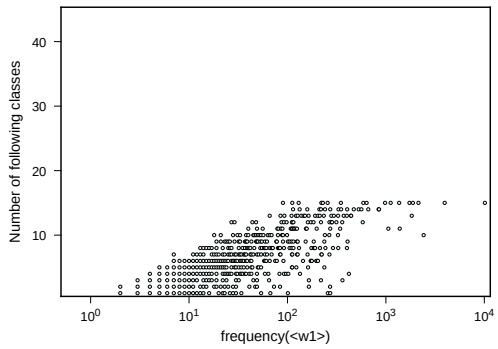


Figure 3.32: Scatter plot of frequency of segment against the number of different categories which that segment is followed by, for a one word segment and a four-bit classification system.

The classification system used in previous experiments was re-described in terms of a binary tree; this is an unbalanced tree, 9 bits at its deepest and 4 at its most shallow. Words were then re-described in terms of the class they belonged to, where a classification could be imposed simply by specifying the bit depth. This experiment takes a bit depth of 4, implying a system with 16 classes, compared to the original system which has just under sixty classes.

N_c against f

Figure 3.32 shows how the frequency of segments of length 1 vary with the number of different following classes in the database. Figures for lengths two to six were also obtained, but are not reproduced here — they follow a predictable pattern. The most interesting feature of the reduced category figure compared to figure 3.10 is that the top half of the plot is squashed rather tha2n being a miniaturised version of the full category figure. This confirms our belief that class-based language models which use coarse-grained classifications should have more reliable (but less informative) segment probability estimations.

Average N_c against Length

An analysis was performed upon these data similar to the full classification experiment. The results graph is displayed in figure 3.33. When this is compared to the equivalent graph for the full classification system, it is noted that the average N_c value is smaller in all six lengths. This result was expected, given that there is a smaller number of possible classes which can follow any given segment.

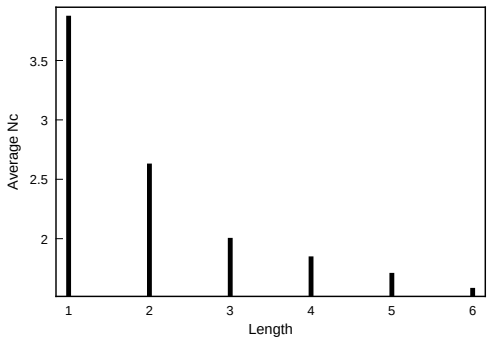


Figure 3.33: Histogram of average N_c against the length, for a four-bit classification system.

3.5.4 Experiments with word-class hybrid segments

Using the standard weighted average language model perplexity results as a baseline, it is possible to calculate the perplexities of different language models which incorporate class information in their individual probability estimations. The following two experiments use the WC-database, using the linguistic classification system described earlier. The nature of this database implies that the sparse data problem will not be significantly improved upon — this is because the context part of the language model, that is, the part which is doing the prediction, is still fully word-based, and words are still prone to the sparse data problem. Still, it allows us to look at the traditionally weak part of a class based language model and compare two versions of it; we can use the better of these two models in future experiments.

Experiment A : Equiprobable Words from Classes

The original language model has as its central component the probability estimation

$$\hat{P}(w_1^n|w_1^{n-1})$$

These two experiments have as their central component the probability estimation

$$\hat{P}(w_1^{n-1}C_n|w_1^{n-1}) \times \hat{P}(w_n|C_n)$$

The first part of this estimation is common to experiment A and B, and corresponds to the weighted average value returned from the hybrid database. The second part of the estimation — the probability of a particular word given a particular class — can be estimated in two ways. In experiment A, the simplest model is suggested; this is one where any word which is a member of the given class is just as likely as any other. More formally, the probability of a word given a class is equal to the inverse of the size of the class.

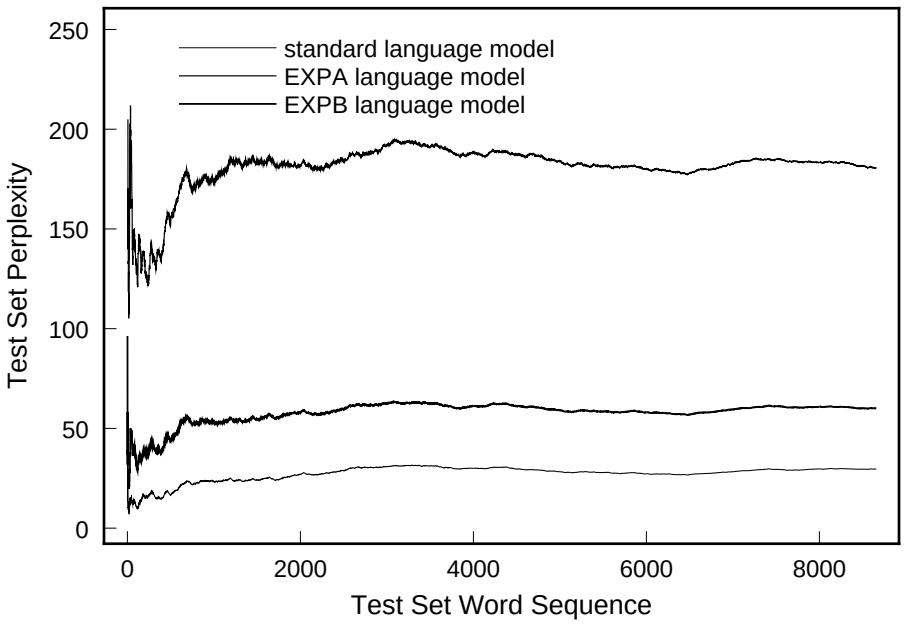


Figure 3.34: Comparison of two ways of estimating the probability of a word, given a class, with the baseline result representing the all word model.

Experiment B : Maximally Likely Words From Classes

A more sophisticated assumption can be made with regard to the likelihood of a word given a class. Instead of the high-entropy

$$\frac{1}{|C|}$$

value, the word probability can be estimated as

$$\frac{f(w)}{f(C)}$$

which produces higher probabilities for words which are more frequent.

Results

The results of these two experiments are summarised in figure 3.34. The most obvious feature is that the informationally weak first experiment increases the perplexity tenfold. The second experiment performs worse than the raw language model, and significantly worse. There are at least two factors which might help to explain why this is so.

First, the estimate of the word probability given a class is context independent. In other words, given the class 'at0', for example, then the probability of 'the' will always and in every case be equal to $\frac{f(the)}{f(at0)}$. Since the underlying principle of language modelling is that the more statistically significant the context, the lower the entropy tends to be with respect

to prediction of the next word, it can be assumed that one of the effects of incorporating a context independent element into the probability estimation is a tendency to increase the entropy. In other words, a system based on only the model used in experiment B will perform less well than the standard language model.

3.5.5 Binary Classifications

For the classification system which has been used in the previous experiments, if it is possible to decrease the class granularity, then the language model which equates

$$\hat{P}(w_1^n|w_1^{n-1})$$

to

$$\hat{P}(w_1^{n-1}C|w_1^{n-1}) \times \hat{P}(w^n|C)$$

would gradually decrease in performance as the granularity decreased. In order to illustrate this hypothesis, a binary tree was constructed which reflected most of the major distinctions in the given classification system. The structure of the tree is displayed in figure 3.35. For any given classification label — for example 'at0' — there corresponds a schema which can be considered as a part declaration of a binary number; in the case of 'at0', this corresponds to the five bits '10010'. By re-describing the system in a semantically relevant way, easy access is provided whenever, for example, the word ⟨`the`⟩ is to be classified as a general ⟨`determiner`⟩; this is done by taking as relevant only the first four bits – in this case, the pattern '1001'. Thus, by varying the amount of bits which are considered significant, larger and larger sub-parts of the tree correspond to more and more coarse classifications of the words in V. There are many arbitrary decsiions made in the manual construction of this classification tree. A theoretically appealing extension of the idea of a binary classification tree is described in section 3.6.

Results

Figure 3.36 shows the results of sixteen independent experiments which estimate the test set perplexity, using the weighted average language model described earlier. Whenever the classification extends so far that each class contains one and only one word, then the language model is behaving like an all-word based language model. Consequently, it gives a perplexity score which matches the all-word model's perplexity score. At the other end of the scale, with a classification of all words into one of two classes — *i.e.* with a classification bit depth of 1 — the perplexity is as high as 170. For the sake of comparison, the manually tagged

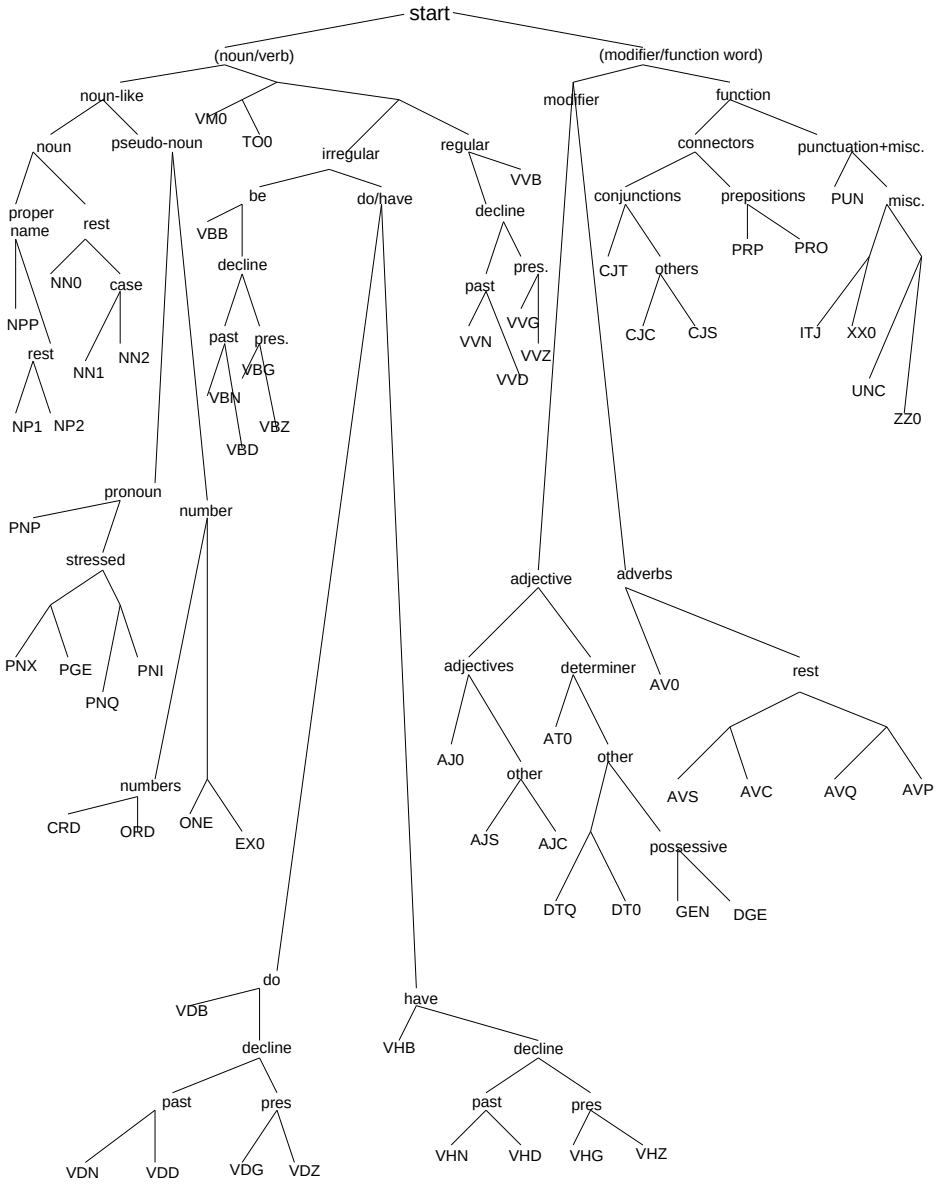


Figure 3.35: Binary tree showing the structure of the classification system.

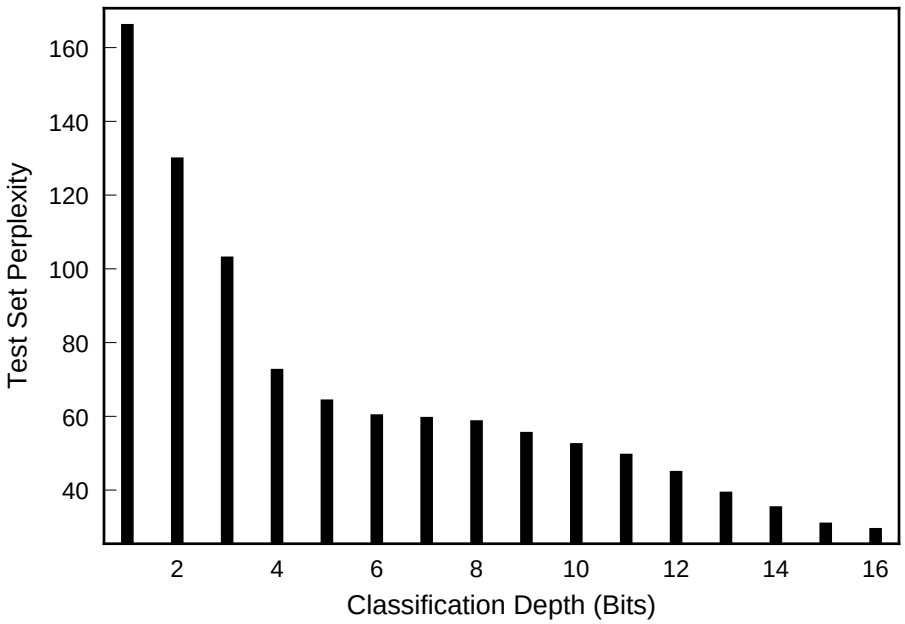


Figure 3.36: Histogram showing how test set perplexity gradually improves as the bit depth, and hence granularity of a classification system increases, for a simple language model described in the text.

classification system gives a perplexity score of approximately 60, which is equivalent to a depth of between 6 and 8 bits.

3.6 Structural Tags

Often, the relationship between a word and its category must be made explicitly — it represents a *relationship* between an object which is a word and an object which is a class. This holds true in implementation terms also — a relationship is maintained between a string which represents the word orthographically, and a string which represents the mnemonic for its class.

At its lowest level, the system accesses a database where, yet again there is a relationship between a word (or class) and its numerical encoding in the database — all words are stored as binary numbers and an object might exist which makes the backwards and forwards mapping between this numerical representation and the more humanly understandable string representation. So, for example, the word ⟨`the`⟩ might correspond to the binary number 1100110000100101 if all numbers are 16 bits long. Clearly, as far as the database is concerned, it is of no consequence for the accessing functions what *particular* number gets assigned to the word, just as long as a few minimum requirements are met — namely that every word is

assigned a unique number.

Next, the idea of a class as a schema is re-introduced (see [65] for a discussion of schemata theory in genetic algorithm research). A schema is a partially or fully defined pattern. It is analogous to a subset in set theory. In this case, we are dealing with binary schemata, which contain two fully specified symbols — '1' and '0' — and one indeterminate symbol — '#'. The '#' can be thought of as a don't care symbol, meaning that the corresponding bit position may be either a '1' or a '0'. We can now re-describe a class as a schema instead of a function relating two distinct object sets. Thus, since the old-style class 'at0' occupies the position '10010' in the binary classification tree of figure 3.35, then we can specify that all words which belong to this class must include the following schema : '10010##########'.

Now, there is no longer any need to store extra information about the class to which words belong, since the very numeric representation of the word contains all of the possible classifications as the set of schemata where bit positions 1, then 1 and 2, then 1, 2 and 3, *etc.* define the particular classification depth being used. These numeric representations of words will be referred to as 'structural tags'.

Also, for a given set of words, V, there can be many possible functions mapping the word into a unique structural tag, each mapping capturing some possibly interesting linguistic feature of the words — for example, their phonetic similarity or their semantic relatedness.

Another advantage of the structural tag system over the traditional word-class relationship system is that, once the tag of a particular word is known, then the system has direct access not only to the immediate class of that word, but also to all of the classes of that word — as many classes as there are bits in the structural tag. This representation has the potential to be incorporated into class-based statistical language processing systems : one reason for using classes was to get frequency information in just those circumstances where the word frequency information was weak. With stuctural tags, the system can calculate the frequency of a segment of words, and if this frequency is not considered to be significant enough to allow it to be used in direct probability estimations, then the word segment can be 'defocused' by treating each of the units in the segment not as words, but as classes of words; the classification can itself continue to be defocused until a segment exists which has an occurrence frequency high enough to give a reliable probability estimation (see figure 3.37).

3.7 Conclusions

In this chapter, we have investigated language models based on the WC hybrid database. We have offered reasons why this might not be the best hybrid combination to pursue —

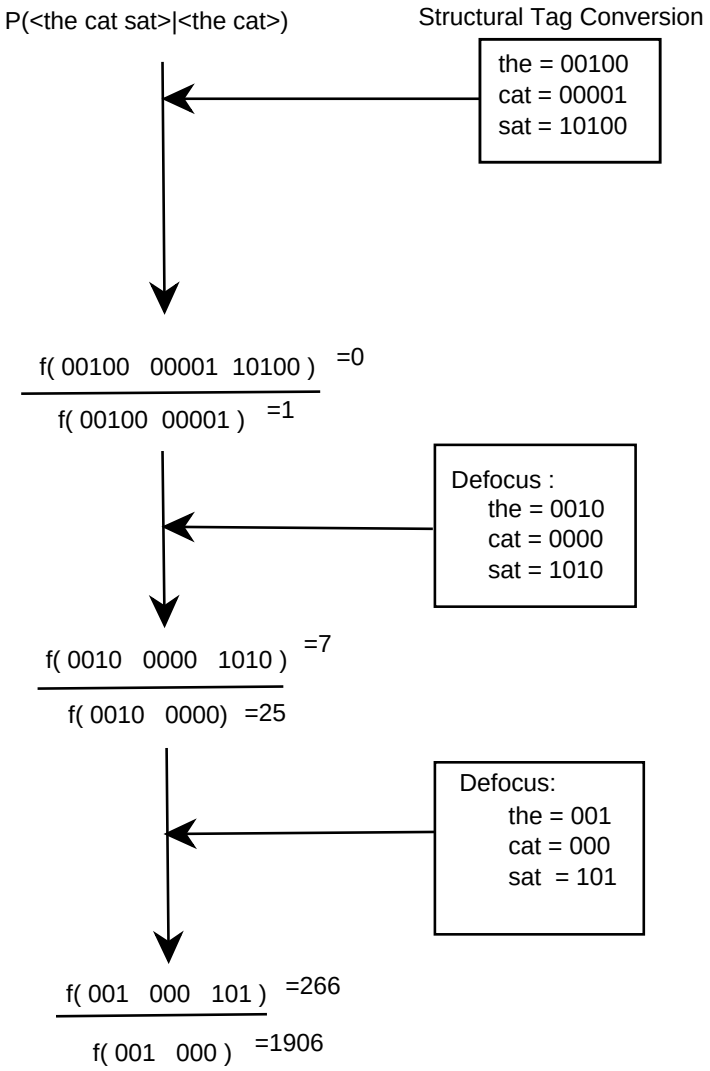


Figure 3.37: Example of defocusing of structural tags. By a process of defocusing, the conditional probability estimate of a segment can be made more reliable, but less specific. This representation could be incorporated into statistical language modelling.

in essence, it is too similar to the WW-database; the extra information provided by the WC database is minimal and it is vulnerable to the sparse data problem in just the same ways as the WW-database.

Secondly, we have performed experiments on a top-down classification system, which required many hours to construct and which does not make many semantic distinctions; the classification is language-specific; the classification is prone to human error; the classification system requires that linguistic tags — usually syntactic — exist for the language being modelled. We would like to find a way to create classifications automatically, in order to avoid these problems.

Thirdly, we have seen that coarse-grained classifications can be useful in language models — they are more reliable but less informative. Ideally, we would like to be able to combine many levels of classification within the one system. The structural tag representation gives us a framework to develop this kind of system.

Finally, we conclude that the maximum likelihood estimate of words given classes is much better than the low-performance equiprobable model. We shall use the maximum likelihood approach to that side of class-based language models in all subsequent experiments. This chapter has contained some useful 'intra-model' insights; in chapter 7, we will incorporate these insights into language model components and test out ways of combining language models — that is, chapter 7 will contain some inter-model investigations. Before that, we shall turn our attention to recent work on automatic word classification.

Chapter 4

Automatic Language Processing

4.1 Overview

For research which aims to investigate how classes might be used in statistical language modelling, the two most important questions are

- How can class information be used in statistical language models?
- What kind of class information can be used in these models?

This chapter contains a review of the underlying themes of recent work on automatic word classification and grammar construction. The first section briefly introduces automatic classification as an example of pattern recognition and the following section describes work already carried out and reported in the literature. There has been a recent confluence of approaches which are inspired by theories in linguistics with those which are inspired by theories in information and probability theory : statistical language models are becoming more linguistically sophisticated and models of language used by linguists are incorporating stochastic methods to allow broader coverage. Finally, there is a short discussion about the underlying similarities between many of the recent word classification systems and a suggestion that the vocabularies of information theory, probability theory and statistics are particularly suited to expressing these connections.

4.2 Introduction

Automatic word classifiers are typically pattern recognition systems [98, 43], the patterns being paradigmatic sets of words related according to some criterion of similarity — for example phonetic, syntactic or semantic. A simple classifier maps input objects to categories.

The objects to be classified (in this case, concatenations of words, phonemes, phrases or individual letters, each representing different levels of analysis of natural language production) can be described in terms of a set of measurable features, usually parts of the object under consideration. If the object has n features, it can be viewed as occupying a unique position in n-dimensional space. A discriminant function partitions this space so that each section represents a category to which that object belongs.

Two types of numeric classifier are deterministic (*e.g.* the K-nearest neighbour classifier and linear classifiers) and statistical (*e.g.* the Bayesian classifier [10, 56]. Chou [29] presents a full description of many common classification systems.

The interdependence of syntagmatic and paradigmatic relations in the description of the structure of natural language is accepted by many language researchers (two recent examples include [148, 52]) — it is this interdependence which leads some researchers to consider the possibility of automatic natural language learning to be remote : in order to construct a system which generates a hierarchical grammar from words, one first needs to know to which categories individual words belong; but to know this, one must have some idea of the positions these words occupy in the grammar. Finch and Chater [52] situate this linguistic problem of breaking into the system of class-definition and rule formation in the wider context of learning new domains generally.

Gillis [63] suggests that, for the problem of language learning from raw data, there are three general types of approach. He first postulates a massive correlation matrix, noting which words appear in which contexts (where context includes morphological features — *e.g.* Resnik [135] — as well as syntactic ones — *e.g.* Schütze [143]). Some psychologists crtitcise this approach whenever it is proposed as a potential model for child language acquisition, but it does not detract from its possible use in constructing speech recognition systems. [1] The other two approaches involve prosodic and semantic bootstrapping. The former might allow the detection of segment boundaries, but it cannot contribute to semantic judgements and conclusions drawn from sentences.

Powers [128] and Brill *et al.* [14] lend support to the approach of structural linguists (*e.g.* Greenberg [69] and Harris [75]) which suggests that a significant amount of the structure which is necessarily inherent within natural language can be detected by formal or statistical means — this structuralist perspective is shared, at least implicitly, by many of the researchers discussed in the present chapter; Tanenhaus [154] discusses their search for discovery proce-

[1] The psychological literature suggests that psychologists have not reached consensus on the relative importance of syntax, semantics and prosody.

dures, which when applied mechanically to a corpus of utterances could, in principle, extract the linguistically relevant units. Liberman [100] notes the increasing interest of the academic community in automatic approaches, especially for those interested in decoding messages in a noisy channel.

Faulk [49] makes the convincing claim that successful human-machine communication using natural language must be preceded by an account of language acquisition which is stochastic and which explains how successful grammatical competence is reached through exposure to a possibly degenerate and certainly finite language sample. His approach is structural linguistic in the sense that he asserts that there is enough structure *in* language for a learner (human or machine) to induce that structure.

Church and Mercer [32] report that the empirical methods introduced in this chapter, described under the noisy channel paradigm, are outperforming the more traditional knowledge-based methods. The case for automatic modelling of linguistic phenomena over manual modelling is made convincingly in Makhoul *et al.* [105]. This paper also discusses some of the practical requirements and problems associated with automatic linguistic modelling. Zernik [163] discusses many of the major limitations of current automatic language processing systems. He presents cogent argument in favour of systems which need minimal human intervention and raw materials which are easily available. This point is especially relevant to the work of Resnik and Bod.

Resnik [135], Derouault and Merialdo [42] and Solomon [147] notice a coming together of information-theoretic and traditional linguistic approaches to language. Resnik's system uses a database of words which have been tagged using a semantic network structure. This database is used as input to a taxonomy-generating system whose principles are based on information theory. He suggests that this synthetic approach models language generation and understanding better than traditional linguistic approaches. He puts less emphasis on the debate over language acquisition but avoids using raw corpora. He claims that lexically based statistics can only generate limited models of language competence. This criticism is discussed in section 8.4. Some researchers are not as sceptical about the quality of the information available from raw corpora — Sampson [139] and Brill *et al.* [15] both present a strong case in favour of distributional analysis. Church [33] also finds surface statistics worth investigating); others chose to make no theoretical challenge to linguistic orthodoxy while using methods which undermine some key tenets of theoretical linguistics (Carroll and Charniak, for example [23] consider phrases 'good' if they occur frequently, regardless of what linguists think of the phrases).

Bod [13] favours models which deal with language performance over competence. He describes four limitations of the competence approach : the problem of ambiguity proliferation, the instability of human grammaticality judgements, the poor facility for modelling language change and the general descriptive inadequacy of all existing rule-based grammars. This last problem has a tendency to become more limiting the larger a linguistically designed grammar gets — the more rules and features, the more chances for inappropriate interactions between them. This insight is analogous to that made by many critics of traditional Artificial Intelligence methodologies [1, 77, 10, 17, 16, 25, 99, 104, 155].

Probability-based procedures for selecting appropriate parses and word classifications possess, in principle, the machinery for improving upon some of these limitations discussed by Bod. The competence grammar of a language user relates to the general structural capacities of that grammar and language, but by itself, it tells us nothing much about the details of how communities of language users have certain linguistic expectations and preferences and how these are used, practically, in disambiguating possibly confusing messages. Bod predicts that the most useful language processing systems will be hybrids of the statistical and formal approaches. The detail of this system is broadly similar to Resnik's work. The main data structure in this case is a tree, whose labels can be syntactic, semantic and pragmatic categories. Thus this work belongs in the same school of approaches to the task of computational language learning as Wolff, Schrepp and Solomon, which variously attempt to derive syntagmatic rules and paradigmatic classes simultaneously.

Studying how much linguistic structure resides in language is an interesting theoretical research topic in itself, though it also has practical advantages over relying on manually constructed corpora. First, and most obviously, running an algorithm on raw text to generate word classes is time and resource efficient — manually tagged corpora are expensive to make, largely because the humans who make the word class judgements do so slowly. Secondly, manual tagging is not a language-independent process whereas the same automatic word classification system could be applied to any language — even one whose syntax and possibly even whose semantics were unknown to investigators. Finally, in some cases, automatic word classification is the only method available to the researcher. This may be true, for example, if the research involves massive corpus size; no truly large-scale tagged corpus exists presently, though the British National Corpus is due for release soon, after several years worth of work. More and more commonly, automatic methods are being used as first steps in the tagging of corpora.

4.3 Automatic Linguistic Processing Methods

In this section, different approaches to automatic word classification are described and connections made between these approaches. The fundamental concept shared by these approaches is one of context. This is the traditional tool of the structural linguist and has its modern origins in the work of Saussure [39, 40]). Another common theme is the idea of distributed representation. Connectionist researchers (for example, Elman [46]) regard this type of representation as vital for the sorts of results which they achieve with their architectures, but it is also present in more traditional frequency-based statistical models. In order to make explicit the nature of what is being represented by a connectionist architecture, statistical clustering techniques such as principal components analysis are sometimes used. With traditional statistical models, equivalent methods exist to make explicit the structure which is distributed throughout the segment-frequency values.

Resnik [135] suggests that there are three broad strands to work on automatic classification : smoothing methods, proximity methods and vector representation methods. In particular systems, one or more of these strands is present. Smoothing methods include models which interpolate unigram, bigram and trigram word frequency information. Generally, they incorporate multiple information sources. If a word is thought of as a point in word space, then the averaging which takes place whenever an interpolated language model is operating corresponds to an implicit classification — those words near the current one can be said to belong to that class. The work of Jelinek [84] and Bahl [5] contains smoothing methods and implicit classification.

Proximity methods use context (operationally defined in terms of surface-distributional lexical co-occurrence statistics) to compute some similarity metric. Confusion matrices and the automatic classification of words described in Brown *et al.* [18], Pereira *et al.* [123] and Brill *et al.* [14] are examples of this method. Resnik [135] uses higher level syntactic and semantic features as the basis of a definition of context.

Vector representations capture the context of a word by a dimensionality reduction. For example, if there are n words which could occur as a context for a given word w, then vector representations try to reduce the dimensionality below n. Many connectionist models of language use embody examples of this method : Elman [45] and Schütze [143] describe two particularly interesting systems. This work, like that of Basili *et al.* [9, 8] concentrates on discovering paradigmatic classes.

4.3.1 Word Association Norms and Mutual Information

Church and Hanks [34] have recently introduced the psycholinguistic term *word association* into the vocabulary of computational linguistics. In psycholinguistics the term refers to the usually semantic priming which occurs between pairs of words. Tests can be performed which measure the lexical retrieval speed for a word like 'doctor'; these tests can be repeated when the subject has been primed by being shown the word 'nurse', for example. Those primer words which lower retrieval time are said to have a high word association index. Also, syntactically close words can act as primers — for example, between certain verbs and prepositions. These association indices are estimated through psycholinguistic experiments with many subjects. Some results of work carried out by Miller and Charles [112] shows that semantic similarity and frequency are inversely related.

The insight that semantic and syntactic relations could be induced from the linear structure of natural language utterances has also been one of the key tenets of structural linguistics [69, 75, 76, 103, 154], crystallised by Firth [54] as follows : "You shall know a word by the company it keeps." Brill *et al.* [14] reiterate the concept in a way which is more operationally useful by describing the various properties of a word as *features* and by claiming that these features license the distributional behaviour of words. Using information theory, the Firthian slogan can be extended : "The degree to which a word is known depends on the degree to which one knows its company" — where knowledge is probabilistic. In other words, useful models of word context can lead to low entropy word prediction systems.

Only recently have computational resources of sufficient power and corpora of sufficient size been made available to the research community [32, 100] in order to perform some of the many experiments implied by the structuralist perspective.

Word associations can be extracted from corpora by borrowing the information theoretic measure of *mutual information* [84, 86, 48, 37]; if $P(x)$ and $P(y)$ are the independent probabilities of events x and y, then the mutual information, $I(x, y)$ is

$$I(x, y) = \log \frac{P(x, y)}{P(x)P(y)} \tag{4.1}$$

This measure compares how likely x and y are to occur together — in the case of words, this means serial occurrence, so that $I(x, y)$ is not necessarily the same as $I(y, x)$ — with how likely they are to occur independently. Finch and Chater [52] and Hughes [79] also use a measure which is independent of absolute frequencies — in the former's neural implementation, this is achieved by using Hebbian learning where the weights are normalised. The higher the likelihood of their co-occurrence, the larger the mutual information value. Church and Hanks

describe some initial analyses of corpora using mutual information. Most of the results of this paper are close analyses of particular word relations and syntactic constructions, reflecting the authors' interest in lexicography.

Brown *et.al* [18] refer to word pairs with high mutual information as *sticky pairs*. Their analysis of large corpora identifies sticky pairs such as the following :

```
Humpty Dumpty
Klux Klan
Ku Klux
Taj Mahal
Pontius Pilate
jiggery pokery
mumbo jumbo
```

This discovery could be put to use in the following way : these word pairs — and by extension word n-grams — can be thought of as word-like, since they nearly always and only occur together. Whenever automatic word classification algorithms are being applied, and they are faced with the task of deciding what to do with the words ⟨`ku`⟩, ⟨`klux`⟩ and ⟨`klan`⟩ for example, instead to trying to group ⟨`ku`⟩ and ⟨`klux`⟩ as types of modifier, the whole segment ⟨`ku klux klan`⟩ should be grouped with noun-like words.

A Mutual Information Example

An experiment was carried out in which the word-bigram mutual information was calculated for any bigram combination of the words ⟨`the`⟩, ⟨`next`⟩ and ⟨`one`⟩ for which there was bigram information. In these examples, all probabilities are maximum likelihood probabilities and logarithms are to base ten. The results are shown in table 4.1 [2]. It is clear from this table of values that the mutual information is identifying differences in word distribution which could be linked to a syntactic distribution. Values which are greater than zero indicate bigrams whose occurrence is surprising if it were the case that all words are being output independently of each other. Conversely, negative values indicate that the output of one word makes some other words less likely. Mutual information can be more useful than simply disconfirming a hypothesis that words are not output with independent probabilities. A high mutual information value between words measures the contribution made by the first

[2] The values given are not quite mutual information values; certain constants have been left out to speed up computation. More details are given in section 5.2.1.

Bigram	Mutual Info.
next one	1.6720
the next	1.3979
the one	0.3095
one next	-0.1474
one the	-0.3272
one one	-0.3720
next the	-0.5797
the the	-0.9936

Table 4.1: Mutual Information values of combinations of the words ⟨`the`⟩, ⟨`next`⟩ and ⟨`one`⟩. The bigram ⟨`next next`⟩ does not occur in the VODIS corpus.

word in reducing the entropy of the second. The values in table 4.1 are all fairly close to zero, which indicates that none of them are particularly sticky. In the case of the bigram ⟨`next next`⟩, where there are no occurrences in the VODIS database, it is safest to assume nothing at all about any effect the first word might have on the second. This is done by assigning a neutral zero value for its mutual information. In a large corpus, this is less than satisfactory, since the bigram's non-occurrence could be interpreted as significant. This becomes more unsatisfactory as the independent probabilities of the unigram words involved become larger. For example, in the VODIS corpus, the words ⟨`the`⟩ and ⟨`of`⟩ occur 1997 and 473 times respectively. Yet while the bigram ⟨`of the`⟩ occurs 83 times, ⟨`the of`⟩ doesn't occur at all. Its mutual information value is set to zero, when a negative value would have been more appropriate.

Sticky Pairs in VODIS

An experiment was carried out which used the concepts of mutual information and sticky pairs on the VODIS corpus, in order to find out which word pairs had high mutual information. The algorithm selects bigrams with the highest mutual information value, only if those bigrams occur with a higher than minimal frequency. Various threshold settings were tried, and the most interesting results are given below.

Sticky Pairs in VODIS : Results

The set of word pairs with high mutual information, derived from the VODIS corpus can be described by six different classes of pair. The nature of these classes is not significant —

Place Names	Discourse	Times	Tickets/Prices	Geography	Customer
Bury St	British Rail	Half past	60 each	round trip	three adults
St Edmunds	Rail Enquiries	How long	70p each	outward journey	two adults
Kings Cross	enquiry office	Quarter past	A third	coming back	under 16
Liverpool Street	Station Manager	Round about	Europe Family	To Nottingham	year old
Fort William	Good afternoon	Same day	Family Card	Via London	how old
Manchester Vic	Great !	an hour	First class	leaving Portsmouth	child goes
Melton Mowbray	Ha !	bit late	Rail Cards	across country	
Milton Keynes	! Ha	earlier than	Rail Europe	cross country	
New Street	very helpful	early start	Senior Citizens	change stations	
St Albans	very much	every hour	an extra	come home	
West Morling	your pardon	hourly service	an ordinary	coming back	
Wickham Market	ever so	minutes late	cheaper than	connect with	
Rowlands Castle	fair enough	minutes past	cheapest way	straight through	
	much indeed	other days	each way	west coast	
	I'm afraid	past eight	family rail		
	find out	past five	first class		
	as far	past four	gone up		
	far as	past seven	monthly return		
	wonder if	past twelve	more expensive		
		quarter past	rail card		
		regular service	rail cards		
		too early	second class		
		too late	third off		
		how many			

Table 4.2: List of the word bigram pairs from VODIS with the highest Mutual Information values — only bigrams with a frequency greater than three were considered.

classes are picked to show certain semantic and discourse related aspects of the conversations found in the VODIS corpus and they account for the majority of the pairs produced by this experiment. The classes relate to place names, discourse establishment — which contains the common connective tissue of many natural language discourses, especially enquiry-based ones — times, tickets and prices, geographical relations and, finally, customer details. The results are shown in table 4.2.

This division is similar to one made by Stephens and Beattie [151], who used a telephone-based enquiries corpus for research into the identification of features which distinguish turn-final from turn-medial utterances. Their four topic areas related to time, cost, route/connection and station services. Their classification was a semantic one, hence it leaves out function-word classes. The present classification includes a distinction between geographical relations

and place names, as well as two discourse-related classes.

Discussion

It is clear that, even with this very simple experiment, and only using bigram statistics, mutual information is a measure of co-occurrence which is powerful enough to discover interesting structural detail. It must be assumed, however, that the particular set of pairs described above are only considered sticky with respect to the VODIS corpus — words like ⟨Kings⟩, ⟨Victoria⟩, ⟨Castle⟩, ⟨Fort⟩ and ⟨Liverpool⟩ will appear in many more varied contexts than they are represented in VODIS. This issue is related to the idiosyncratic domain of the corpus, but also to its relatively small size. The potential limitation will become important whenever an automatic word classification algorithm is applied to VODIS, since this algorithm will discover sets of similar contexts for word groupings, which do not exist in free discourse. While the precise six-class taxonomy described above is arbitrary, it does describe commonalities among word pairs with high mutual information values. Stephens and Beattie classified utterances separately, yet achieved 97 % agreement between themselves. This is not surprising, since it reflects the fact that the verbal behaviour associated with telephone enquiries tends to centre around a relatively specific set of objectives. This type of *low entropy* behaviour is what anthropologists try to describe whenever they do ethnography. Very low entropy behaviour becomes, by definition, less and less information-bearing. Anthropologists describe such behaviours as ritual behaviours. For example, the verbal behaviour of people at a church service, the priest and his congregation, is very low entropy, ritualised linguistic behaviour.

Many other experiments can be carried out using non-adjacent bigram information as well as adjacent bigram information. Brown *et al.* [18] and Schütze [143] discuss these modifications.

4.3.2 Discovering Constituent Boundaries with Mutual Information — The Shannon Game

Language is made of constituents. Grammar determines the rules of combination of these constituents. In any segment of text, there are many ways of sub-dividing that text. From a language processing perspective, the most interesting way is by dividing it into the constituents of the grammar which generates that language. Constituents are hierarchical — at one level, the text is divided into sentences, at another, into sub-sentential constituents. Mutual information can in principle be used to look at all possible sub-divisions of a text

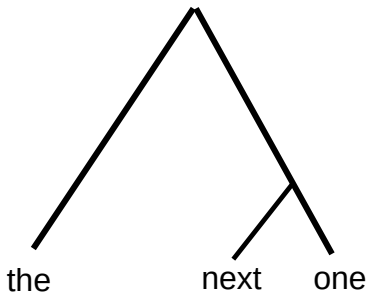


Figure 4.1: Automatically generated parse of an example test phrase — ⟨`the next one`⟩, based on Mutual Information statistics calculated from the VODIS corpus.

and to identify that set of divisions which best corresponds to a grammatical parse. It can do this by identifying constituent boundaries.

For example, if an automatic parser received the segment ⟨`next one`⟩, then it could use the mutual information values shown in table 4.1 to create a parse for the segment. Figure 4.1 shows the parse which would result, corresponding to a grammatical parse of a noun phrase into determiner, adjective and noun.

Brill *et al.* [14] and Church [33] develop the use of mutual information within computational linguistics in several interesting ways. They apply the concepts to grammar induction and automatic parsing. They use a corpus which has been annotated with parts of speech and develop a constituent boundary parsing algorithm which takes a sentence from the Brown corpus, such as

```
He directed the cortege of autos to the dunes near Santa Monica
```

and parses it as follows :

```
(He(directed((the cortege)(of autos)))((to(the dunes))(near Santa Monica)))
```

In other words, they use mutual information and word classes to produce robust parses of sentences. If it can be demonstrated that mutual information can do the word classification *as well* then it offers the computational linguist in a single tool something which can derive word classes and use these classes to parse the sentences of huge corpora — all automatically, relying on the statistics of words and constructions which have been attested in a corpus.

The Brill *et al.* parsing algorithm uses mutual information *minima* to discover constituent boundaries; the same insight has been applied in the work of Wolff, using a rule-based system [159, 160], by Elman [45, 47], Pollack [126], Jordan [88], and Gasser [62], using connectionist architectures and by Faulk [49], directly using Harris' idea of a *variety index* value

being minimised at constituent boundaries It is also related to the idea of discovering useful schemata with genetic algorithms [65, 77]. The same periodic rise and fall of uncertainty is described using information theoretic terminology by Shannon [144]. Attneave [4] describes a similar phenomenon in the simulation of visual perception — information is highest along contours and boundaries in a visual image, and highest of all when the rate of change of the boundary is highest.

In Brill *et al.*'s formulation, word class n-grams are examined in order to find likely constituent boundaries, or *distituents*. Their hypothesis states that a form of mutual information called *generalised mutual information* will be able to identify distituents. This can be explained further by an example.

If the class n-gram is ⟨`det noun verb`⟩ and we are looking for the most likely constituent boundary — *e.g.* if the best partial parse is `((det noun) verb)` — then the probability $P(\langle$`det noun`$\rangle)$ should be significantly higher than $P(\langle$`det noun verb`$\rangle)$. Informally, this captures the intuition that good constituents should occur in many contexts. In terms of the task of predicting which class comes next, the entropy should remain low, and possibly even get lower, until the constituent ends, at which point the entropy for the next class should be significantly higher. These high entropy break-points in effect mark off the structure of particular utterances.

This method leads to successful parse tree estimations for a test set of unconstrained free discourse.[3] The authors also tackle the problem of automatic word classification, using an algorithm which is similar to one used in Brown *et al.*, described below.

4.3.3 Generating Word Classes Entirely From Statistics

In the work cited above, Brown *et al.* apply an algorithm which induces classes for words. Initially, every word is assigned its own class. Next, a pair of classes is merged. In order to decide which class pair is to be merged, each class pair in turn is merged, and the average mutual information for the new set of classes is calculated. The merged class pair which minimises the loss in average class mutual information is the one finally selected. This process is carried on until the desired number of classes have been distilled. The work is incorporated into an interpolated language model and successfully lowers the perplexity. More interestingly, a language model based on *only* these classes leads to a perplexity which is only slightly higher than the all word perplexity. The benefit being that the class based system takes up much

[3] For sentences of length less than fifteen, the parser averages two errors per sentence, rising to between five and six errors for sentences between sixteen and thirty words long.

less storage space — approximately one third of the all word system, according to Brown *et al.* Of even more importance is the fact that such analysis can be carried out; for linguists, and especially for computational linguists and others who are concerned with the observed structure of language, it provides an empirical opportunity to test out the robustness of the largely theoretical constructions with which they have been working. Unfortunately, the problem of finding that series of class merges which maximises the whole class average mutual information has not been solved. Brown *et al.*'s heuristic approach — of locally maximising average class mutual information – does however lead to very interesting results. Their algorithm discovered, among others, the classes :

```
'Friday Monday Thursday Wednesday Tuesday Saturday
 Sunday weekends Sundays Saturdays'
'mother wife father son husband brother
 daughter sister boss uncle'
'had hadn't hath would've could've should've must've might've'
'head body hands eyes voice arm seat eye hair mouth'
'feet miles pounds degrees inches barrels tons acres meters bytes'
```

These classes demonstrate the power of mutual information when it is combined with enormous computational resources and a 365,893,263 word training set and support the claims made for faster, bigger computers and larger corpora, made in Makhoul *et al.* [105].

Brill *et al.* (in the same paper — [14]) report their attempts to discover the word classes of a language. Their approach is similar to that of Brown *et al.* — they attempt to use a distributional analysis based on word co-occurrences to cluster classes of words — but the algorithm they implement is not based on average class mutual information. Instead, they describe the requirements for two words to belong to the same word class in set-theoretic terms. Two words, x and y belong to the same class if and only if word y contains all of the features of word x and word x contains all of the same features of word y. The features of x are operationally described in terms of the set of bigrams where x is one of the words. A second set can be generated from this set of the features of x by replacing x with y. If this new set of bigrams is attested in the original training corpus of bigrams — that is, if x can be replaced by y in all of its contexts and the resulting segments are still attested, then it can be concluded that y has at least all of the contexts of x. The reverse process should show that x has at least all of the contexts of y. These two results imply that they share the *same* set of contexts and hence can be classified similarly.

The previous description can be extended by bringing in probability theory. Instead of the all-or-nothing decision that two words share the same context, probabilities are calculated which determine the degree of similarity, and hence the degree of closeness of two words.

4.3.4 Neural Networks — Word Classification using Mutual Information analogues

Recurrent connectionist architectures [46, 47, 45, 88] have been applied to the task of the discovery of the structure of language from its serial expression. In these cases simplified grammars are used, with restricted vocabularies. Connectionist researchers also recognise that high entropy break points represent useful ways to proceed with the discovery of linguistic structure. Elman [45] uses slightly different terminology — he notices that time-varying error signals in recurrent nets can provide clues to structure. One of the challenges which he hopes his work addresses is the Chomskyan claim [28] that adequate descriptions of language cannot be generated *only* from a linear sequence of words; Elman attempts to show that an explicit higher level structure, capturing the paradigmatic grammaticality relations is not necessary, since this can be induced from the linear word stream.

In a series of more and more sophisticated experiments, Elman demonstrates how recurrent neural networks can be trained to predict which element in a stream of elements comes next, whether these elements are bits, letters or words. He demonstrates how the network successfully learns the XOR prediction problem, how it breaks continuous streams of letters into word-like parts and even how syntactic and semantic class information is induced from word streams. This experiment will be described in much greater detail later. His most important insight was to recognise that after a network has been trained up to predict which word might come next in a continuous flow of words from sentences generated by a simple grammar, the network's hidden nodes must be representing, in a distributed way, the syntactic-semantic distinctions of that language. Elman performed cluster analysis on the hidden units in order to discover the nature of this induced structure. Two aspects of this system make it difficult to use in statistical language modelling — first, the classes are not explicitly available and second, the quality of results when scaled up to real language data has been challenged [131].

Reilly [133, 134] develops this work by training another neural network to take as input the hidden layer activation state and to output a partial but explicit parse. Thus, in his system, just as in Brill *et al.*, algorithms exist which can automatically parse incoming streams of words, using the structure which is implicit in language, but without having an explicit

traditional linguistic component. Brown *et al.*'s class-based n-gram language model can also be described as a stochastic grammar which is implicit in the frequency statistics, and also distributed. From this perspective, traditional grammars can be re-described from within the stochastic grammar framework, but where the n-grams correspond to sentence boundaries and where the probabilities are clamped to 1 or 0. This clamping corresponds to the *competence* idealisation introduced by Chomsky [28], leading to an undervaluation of preference based constraints.

Kohonen (described in [10]) developed *self-organising* networks, unsupervised learning systems which create their own classifications directly from the training set, by a method of vector quantisation. Multi-dimensional data is mapped onto lower dimensional space. Vectors that are located in a similar area of feature space will be classified correctly even though the network has not been presented with them in its learning phase. Kohonen uses a modified Euclidian distance metric to compare vectors. He also introduces *learning vector quantisation*, which is the supervised equivalent to a self-organising map. This process can form part of a feedback system which fine-tunes the network's classification system. He has embedded this network in a larger system which performs as a phonetic typewriter. The whole system, from human voice input through microphones, to the final orthographic production of written Finnish has an accuracy range of 92 and 97 percent, for unlimited vocabulary. It is also flexible in that it can be trained (in under ten minutes) to respond to new users — using learning vector quantisation.

Scholtes [142] implements his *Data Oriented Parsing* system with a Kohonen feature map. The system uses structural features together with statistical information from a corpus. His parser produces a set of ranked parses for an ambiguous sample sentence; it produces partial parses for incomplete sentences, wrong sentences and new sentences which contain several totally unseen words or structures.

Schütze [143] describes a hybrid system which uses statistical clustering methods like truncated group average agglomeration along with a bidirectional recurrent network. He claims that his work can be applied efficiently to very large vocabularies; his system achieves impressive automatic word classification and part-of-speech labelling. One reason why this might be is because he includes in his operational definition of context not only contiguous bigrams – that is, bigrams of the form $\langle w_{i-1}, w_i \rangle$ and $\langle w_i, w_{i+1} \rangle$ but also preceeding and following bigrams at a one word distance — *i.e.* bigrams of the form $\langle w_{i-2}, w_i \rangle$ and $\langle w_i, w_{i+2} \rangle$. Finch and Chater [50], Hughes and Atwell [80] and Brown *et al.* [19] use this extended conception of bigram statistics.

Schütze takes the 5,000 most frequent words from a large corpus and generates a 5000 by 4 sparse matrix containing the relevant bigram frequencies. This matrix is passed to a sparse matrix algorithm which implements a singular value decomposition. This produces a 15-dimensional real-valued matrix for each word preserving similarities between words.

Wyard *et al.* [161] have designed a *single layer higher order neural net* which takes as input tuples of word class units — *e.g.* ⟨`adj noun`⟩ — and tries to identify those tuples which help in grammaticality decisions. The input sentences are either grammatical or not, and the net is trained, using a form of punishment learning, to output a simple binary grammaticality decision. This system could be used as a pre-parse filter. Positive sentence samples are generated from a context free grammar and negative samples are randomly generated. The neural net was also supplied with extra information which made sentence boundary determination trivial. This contrasts with Elman, whose net has as input a constant stream of words with no explicit sentence boundaries.

Neural network implementations have the advantage of being more cognitively plausible and allow prediction and testing as well as various manipulations of the network, which might be relevant for explorations of language disfunction.

4.3.5 Statistical Clustering and the Replacement Test

One interesting statistical model is that of Finch and Chater [51, 50, 52]. They carry out a simple statistical measure on a corpus and use this to derive syntactic and semantic categories — their work considers these categories as different in degree only, and not in type.

They derive their similarity metric from a consideration of the 'replacement test' of theoretical linguistics, which suggests that lexical items which are distributed similarly should receive similar linguistic categorisations. If

$$\langle C, w \rangle$$

is a well-formed sentence in a language, where C is the set of all contexts and w is a particular lexical item, then w and w' are said to belong to the same class if

$$\langle C, w' \rangle$$

is also well-formed. Since the notion of well-formedness is not simply incorporated into statistical natural language processing systems, Finch and Chater have decided to define the context of an item to be the two words either side of the word. They use the Spearman Rank Correlation Coefficient and cluster analysis then places words of similar distribution close

to each other in a dendrogram. Other classifier methods can be applied to the distribution results — the 10-nearest neighbours of the item ⟨`three`⟩ are as follows : three, four, five six, several, real, black, old, high, local, white. And for ⟨`I`⟩, they found : I, we, they, he, she, you, I've, doesn't, don't, I'm, didn't. Their approach is based upon ideas similar to those discussed in section 4.3.3.

The clustering results proper are impressive, even for bigram statistics [50] — they are based on a corpus of 45,000,000 words. Later experiments use associative networks with Hebbian learning to calculate the distributional statistics.

One of their experiments involves the four positions (two either side of the word in question) as four vectors, each of 150 dimensions, corresponding to the frequency of the 150 most common words in the corpus. These four vectors define a simplified and computationally tractable operational definition of context. For every word, w_i, a cluster of 600-dimensional points is calculated; the Spearman's Rank Correlation Coefficient is calculated between one word's context and another's.[4] Finch and Chater have also begun to apply some ideas from self-organising neural networks (Kohonen's work) to their own statistical bigram model, with slightly less successful results.

This statistical and neural approaches suggest an underlying connection between both. Also, the 'replacement test' involves ideas which are obviously similar to the more general approach of estimating mutual information statistics, and to the 'variety index' of Faulk [49]; it is also similar to Brill *et al.*'s distributional analyses and Schütze's hybrid neural net and statistical clustering approach.

Once a set of word classes have been induced, these classes can be used to induce grammatical rules, which in turn can be used to improve the original classification.

4.3.6 Formal Language Theory

This is a form of *syntactic pattern recognition* [66] where input vectors are considered to be well-formed expressions of some abstract grammar G_i. Naumann and Schrepp [115] suggest one method of inductive learning of a grammar which will parse a given corpus. They use an incremental learning algorithm which produces a sequence of grammars, each of which parses

[4] Spearman's Rank Correlation Coefficient is a non-parametric measure of the association between two variables x and y, when the distribution of x or y (or both) cannot reliably be assumed to be normal.

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n^3 - n}$$

where d_i is the difference between the rank of the ith x value and the rank if the ith y value, where the n values of x and y have been separately arranged in ascending order.

the corpus more and more successfully. New sentences from the corpus are parsed to produce partial structural descriptions; a set of new grammars for the corpus is created which will parse the sentences in question, and the grammar which makes the smallest inductive leap is picked to be the new grammar — in other words, the grammar which has the smallest maximal set of productions is selected. This process is continued until the corpus can be adequately parsed. The main disadvantage with this approach is the danger of over-generalisation. Input to formal language grammar inducers is usually in the form of grammatically well-formed sentences, making these systems less plausible. Systems which do not use probabilities to guide selectional preference find modelling language acquisition more difficult : in effect, all grammatical sentences are considered equiprobable.

Work on formal languages as models of natural language follows from work by Chomsky [27], Gold [64] and more recently work by Berwick [12] on application of the Subset Principle — a technique which, while restricting itself to positive-only input, guesses the narrowest possible language compatible with the data given so far. This principle could explain the observational evidence reported by Brown [20].

4.3.7 Categorial Grammar

Solomon and McGee-Wood [148, 147] have applied categorial grammar [108] to the task of learning a grammar. The process is semi-automatic and uses as its corpus a sample of childrens' utterances.

In categorial grammar, a small number of atomic categories (usually s for a sentence, np for a noun-phrase and n for a noun) are postulated and all other words are defined in terms of complex categories made up of some combination of these primitives. For example, a word which was an intransitive verb might be described by the functional category $s\backslash np$, which indicates that the word in question is of that category of words which, when prefaced by a noun phrase, produces a sentence. Thus the complex category captures the main syntactic features of the class of intransitive verbs. The process of inducing the grammar, then, is equivalent to that of identifying all of the commonly occurring complex categories, after some of the words in the corpus have been manually tagged. Due to the complex nature of the categories, the resulting lexicon implicitly captures the full richness of grammatical relations. In the system, word ambiguity is dealt with by allowing a word to have multiple categories.

This approach is also structuralist — each complex category represents a location in linguistic space for a word to reside. Since the system can handle ambiguity, then each

category represents an instance hypothesis, the sum of which corresponds to a distributed cluster in the abstract linguistic space. The learning algorithm uses the surface features of a word's distribution as the guide to placing that word in linguistic space.

4.3.8 A Redundancy Exploiting Approach based on Unification

Another researcher, Wolff [159, 160] sees the bootstrapping problem as being solved through data compression techniques. Like Finch and Chater, Wolff exploits the informational redundancy in a corpus in order to develop a grammar which represents that corpus. He extends this idea by suggesting that the development of language in humans is driven by the minimisation of information storage and retrieval. He defines *efficiency* in a body of information as *power / size*, where power is the expressive power of the body of information — *i.e.* the non-redundant information it contains, and size is the number of bits in the body of information. A grammar, then, which codes for a set of utterances, can be considered to have captured the power of the utterances, but is much smaller in size. This gives him a measure to compare different grammars which cover a given set of utterances.

4.3.9 Genetic Algorithms And Classifier Systems

Genetic Algorithms [65, 77, 107] allow learning through a process of natural selection with respect to an optimisation task. A population of estimates of solution hypotheses for a given problem are compared using a fitness function. There are also mechanisms which supply mutation or crossover, or both; these randomise parts of a hypothesis, generating new hypotheses and preventing the learning from settling into local minima.

For a language system, the basic units are segments (initially just direct concatenations of utterance taken from the corpus). These units are not primitive — they have parts which can be analysed (for example, the individual items in the segment, the length of the segment, what type of item is present at what position in the segment). A genetic algorithm could be applied by re-casting the grammar representation in the following way.

A grammar of segments is described where items can be i-units (lexical items), s- or syntagmatic units and finally p- or paradigmatic units. For example

$$\langle the, cat, p, s \rangle$$

is a segment where p might represent the paradigmatic set of verbs; s might represent the syntagum $\langle$ on, the, mat $\rangle$, or perhaps $\langle p_1, p_2, p_3 \rangle$, where p_1,p_2 and p_3 are prepositional, determiner and noun paradigmatic sets respectively. Paradigmatic sets can be defined in

the same way as Wolff [159] and Finch and Chater [51, 50] define them. A syntagmatic transformation can be defined as follows. Given

$$\langle u_1, u_2, \ldots, u_n \rangle$$

and

$$1 \leq x \leq y \leq n$$

then the transformation

$$\langle u_x, \ldots, u_y \rangle \longrightarrow \langle u_s \rangle$$

creates the following segment

$$\langle u_1, u_2, \ldots, u_{x-1}, u_s, u_{y+1}, \ldots, u_n \rangle$$

The segment $\langle i_1, \ldots, i_n \rangle$ is an instance segment, taken directly from the set of all segments of the corpus. A process is defined which produces many non-instance segments from instance segments; at any time, the set of these non-instance segments is said to be a particular grammar. The choice of new non-instance segments is driven by measures like mutual information, or perhaps the structural linguistic replacement test.

A population of grammars $G_1, \ldots, G_p$ is created and a fitness measure $f(G_i)$ defined which takes a random test set of instance segments from the original corpus (seen or unseen) and scores each G_i. At each generation, the top scorers are reproduced into the next generation and, given a fixed population size, the poor scorers die out.

The grammar G_i, over many generations approximates more and more successfully to a grammar which can deal with all of the corpus. Once learned, the segment grammar can provide extra statistical information about any test instance segment.

Also, since G_i should have new (non-instance) segments entering into it at every generation, if this process is unconstrained it will permanently increases the size of the grammar. A size limit could be imposed such that each segment in G_i is in competition with every other. Various measures of segments can be taken — *e.g.* how often that segment is used in an end-of-generation test segment — which can determine the fitness of each segment in G_i. This should allow for many grammatical constructions to be represented, *in theory*, but only those constructions which are regularly used should appear often in practice. By this method, over-generalisation can be minimised. This idea is similar to that of Wolff, who measures a rule's fitness in terms of a maximisation of the product of its frequency and size.

The *building blocks* theory discussed in Goldberg [65] is analogous to maximising mutual information in order to find constituents; *linkage* is the device by which statistically significant

concatenations of bits along the genotype of a potential solution are propagated into further generations, thereby remaining significant. This description shares common features with Brill *et al.*, who try to identify constituents by virtue of their high relative frequency.

The above description could be extended by the use of classifier systems [78, 7, 155] for introductions to the theory of classifier systems). The rule of transition from instance segment to class segment can be seen as a simple if-then rule, which is the atomic element of a classifier system. That set of if-then rules which tends to cover the test segments in the most efficient way are rewarded; unhelpful or wasteful rules receive less reward and tend to die out. When linked to a genetic algorithm, which, through crossover and infrequent mutation, new grammatical sub-strategies are introduced into the fray. The system then tends to evolve towards better and better approximations of the underlying grammar of the tested language. This approach is like an evolutionary version of Wolff's rule-based symbolic system and Schrepp's more formal symbolic system.

Antonisse [2] develops a reformulation of genetic algorithms so that they can represent any problem which can be described as a formal grammar. He does this by re-defining the crossover operator so that the newly created string is well-formed in some grammar. This is achieved by tagging the trailing and leading edge of each split string with a tag which captures the relevant string fragment's position in the phrase structure from which it originally came. Now, two strings can link if and only if their trailing and leading tags can be unified. Koza [94] has also developed work along this direction in his *genetic programming* paradigm, though Antonisse claims that his own work subsumes Koza's.

4.3.10 Classification using a Lexicon

Resnik [135] brings the techniques of information theory to a noun taxonomy which has been constructed by hand in the form of a semantic network. He claims that this helps in elaborating an empirically adequate theory of selectional constraints, which he bases upon the concept of 'preferred association'. Essentially, information theoretic measures provide each word with a 'selectional profile', which can be compared numerically with other such profiles with respect to particular associations.

The model he constructs seems to deal well with the traditional examples of selectional constraint; it also shows useful properties when a class of verbs is analysed with respect to its argument realisation properties; finally, Resnik reports success when the system is used to syntactically disambiguate lexical items in an unconstrained text.

Resnik's work uses the WordNet lexicon, which is described by Beckwith *et al.* [11]; it

has been constructed on principles of human lexical organisation, developed by psycholinguists. It provides a rich representation for language modellers to use, whether they base their models on formal grammar and X-bar theory or on statistical collocations and mutual information. WordNet currently contains information about 64,000 different nouns, verbs and adjectives. It has been developed from a synchronic perspective, rather than the usual diachronic perspective of most dictionaries.

Bod [13] discusses an approach called *data oriented parsing*, where the corpus upon which the language processing system operates consists of what he calls *language experiences*. Typically, the corpus contains many small language fragments, and analysis of a new input corresponds to building the input in terms of these fragments.

Liddy and Paik [101] calculate the correlation between pairs of semantic tags supplied by the LDOCE dictionary using the correlation coefficient statistic. They include the information this provides into a hybrid system which includes heuristics for combining the multiple information sources.

At a higher level of abstraction, the Liddy and Paik system is performing the same sorts of operation towards a goal of successful prediction as statistical language modellers. In their system, the object to be predicted is a rich parse of a given sentence — this parse could include any number of types of semantic and syntactic representation. Towards this goal, they use prior information about the likelihood of encountering sequences of semantics currently hypothesised in their on-going disambiguation. With statistical language modellers, their given data are speech patterns, and their predictive goal is a well-formed, maximally likely sentence stream. Towards this end, they can use word class information; this usually takes the form of syntactic information, but it could also include semantic information. In a similar way, automatic translation fits into this abstracted prediction process — instead of predicting an appropriate semantic structure, translation systems work towards appropriate foreign sentences, though they may include an inter-lingual semantic stage (see [153] as an example and [109] for a discussion of the continuing influence of inter-lingual ideals on cognitive scientists and Artificial Intelligence researchers). This claim about the underlying similarities between work on translation, semantic disambiguation and language modelling is partly supported by the recent use researchers have made of parallel texts : in testing word sense disambiguation models, some researchers [59] have been using examples of text where the semantically ambiguous words are variously translated into two or more distinct foreign words, each of which broadly approximates one of the original word's senses. For particular sentences and particular semantically ambiguous words, the translated text affords an op-

portunity to evaluate the sense chosen by the disambiguation model. This testing method is appealing because it avoids reliance on computationally expensive tagged corpora or lexicons which are sufficiently large. One of the difficulties with this method is an underlying assumption about the semantic importance of translation differences — just because there are three foreign words which approximately translate some word, it cannot be concluded that there are three distinct senses.

Burger and Connolly [22] provide good examples about the danger of estimating statistics of high-level — that is, non-surface — linguistic phenomena : they construct a Bayesian Network partly by hand (designing its overall structure) and partly from frequency information (estimating the arc transition probabilities). They are forced to calculate corpus-derived statistics of events which are the constructs of linguists — for example, 'discourse focus', a phenomenon which is unseen in a raw corpus, and which may be constructed only upon acceptance of a particular linguistic theory.

Some researchers prefer to use hybrid statistical and syntactic models to improve performance in disambiguation, recognition, part of speech tagging [15, 33] or generation. Rohlicek, Chow and Roucos [136] have reported success using a small corpus and a set of manually constructed sentence templates, unto which sentences are mapped. They then use a Markov model, using the reduced number of training parameters which their sentence template system allows them, to construct a useful model of language which can automatically tag words. Basili, Pazienza and Velardi [8] also add in some syntactic and semantic information to improve the performance of their system by making the data more significant. This work extends statistical language modelling in the same direction as Resnik. Basili *et.al* continue their work [9] by developing a verb clustering algorithm which manages to identify semantically plausible classes of verb.

4.4 Commonalities Between Automatic Word Classification Systems

Some of the statistical approaches described in this chapter share common features with each other; they are also related to structure-discovering processes in other areas of cognitive science simulation, most apparently in some research on visual processing. The underlying principles which unite this research come from information theory and statistics.

In the short period after Shannon's description of an information theoretic approach to language processing [144] and before Chomsky's influential criticism of finite models of lan-

guage [27] psychologists investigated some of the powers and weaknesses of the distributional approach [111]. This work has continued, despite its lack of mainstream psycholinguistic appeal [112]. Contemporaneously, psychologists were also investigating the significance of an information-theoretic approach to the visual system; an early discussion of informational visual processing can be found in Attneave [4]. There has been no comparable rejection of information theory from cognitive models of visual processing.

In his article, Attneave spends some energy convincing the reader that the same principles of information theory are being used in language processing; he suggests that visual information is concentrated along contours, which fits well with the idea of entropy as a measure of uncertainty : this has some parallels with the recent idea of a distituent being associated with the point of minimum mutual information, described well in Brill *et al.* [14]; this idea is also supported by the pattern of prediction error rates in recurrent neural networks [45]. Attneave also links the psychological idea of *gestalt* with the information theoretic concept of high redundancy. In linguistics, this corresponds to utterances such as : ⟨`the cat sat the mat`⟩; the preposition is inferred by the hearer in order to construct a meaningful sentence. Next, he states how these principles lead to a questioning of the connection between perception and inductive reasoning : that is, the boundary between perceptual information gathering and the central processing which the brain is supposed to perform. A convincing description of the nature of this boundary has recently been given in Dennett [41]. Attneave's main contribution in this article was to give some heuristic methods for reducing redundancy in an informational field. Many of these ideas can be subsumed by Algorithmic Complexity Theory [24, 149, 93]. Attneave also claims that something similar happens when good science is in operation [5]. This opinion has recently been championed by Finch [53]; it also bears similarities to the philosophy of science of Popper [127].

Some connectionist researchers have described the functioning of their neural systems in terms of *information maximisation*, under certain constraints. Linsker [102] describes a multilayered neural net which uses Hebbian learning to self-organise feature-analysing cells. He states that the organising principle behind the changes in connection strength involves maximising the amount of information preserved in the signal as if moves through the layers. This is equivalent, given certain constraints, to maximising the statistical variance of each layer's output activity. Linsker not only shows that there is a relation between Hebbian and Hopfield networks and information theory, he also demonstrates the mathematical link

[5] "The abstraction of simple homogeneities from a visual field does not appear to be different, in its formal aspects, from the induction of a highly general scientific law from a mass of experimental data" (Attneave [4], p187)

between these connectionist learning rules and statistical variance.

The work of Finch *et al.* uses Spearman's rank correlation coefficient, ρ :

$$\rho = 1 - \frac{6 \sum_{j=1}^{k} D_j^2}{k^3 - k}$$

which can be shown [110] to be equivalent to

$$\rho = \frac{\sum_{j=1}^{k} (X_j - \overline{X})^2 (Y_j - \overline{Y})^2}{\sqrt{[\sum_{j=1}^{k} (X_j - \overline{X}) \sum_{j=1}^{k} (Y_j - \overline{Y})]}}$$

the correlation coefficient between two variables X and Y, which is just the covariance of their standard forms [21]. The closer this statistic is to 1, the stronger the correlation between the two sets of variables. Linsker has shown how covariance maximisation is related to information maximisation in a neural net; a related conclusion is that variable distributions which maximise the Spearman's rank correlation coefficient also maximise information.

Similar links between neural network performance and information processing can be found in Plumbley [125] and Atick *et al.* [3], where the goal is minimisation of redundancy, corresponding to a minimisation of output channel capacity. A connection between information theory and artificial neuronal architectures is made by Gorin *et al* [68], who construct a network the weights of which are defined by mutual information.

Pereira *et al.* [123] cluster words using the Kullback-Leibler distance, or relative entropy. They use it to minimise the information loss in using a class distribution rather than the actual word distribution; mutual information is defined [37] as the relative entropy between the joint distribution, $P(X, Y)$ and the product distribution $P(X)P(Y)$; that is

$$I(X, Y) = D(P(X, Y) \mid\mid P(X)P(Y))$$

where $D(p_1 \mid\mid p_2)$ is the relative entropy between probability distributions p_1 and p_2; relative entropy as a measure of the distance between two distributions is closely related to covariance and the correlation coefficient.

Burger and Connolly [22] construct a Bayesian Network in the form of a tree and use sum squared error minimisation to calculate parameters for their system, which attempts to resolve anaphoric reference. This system is similar to Bahl *et al.*, who use entropy minimisation to build a decision tree for language modelling. Burger and Connolly derive their measure from the gradient descent of back-propagating neural networks, while Bahl *et al.* construct their language model equivalence classes by minimising the average entropy of leaf distributions — that is, they attempt to discover the maximally informative binary question at a tree-node.

Fisher and Riloff [55] use the t-statistic as a measure of co-occurrence likelihood between two items. It too is calculated from corpus frequency information and can indicate strong

correlations between items. Grefenstette [70] suggests that the Jaccard distance similarity measure leads to interesting language collocations. Kneser and Ney [92] and Ney, Essen and Kneser [116] include examples of word classification systems which, while not hierarchically clustered, use an optimisation technique based on decision-directed learning; their optimisation measure is training set perplexity.

4.5 Word Classification Methods — Conclusion

It should be clear from the previous discussion that, though many of the methods of word classification may appear to use different approaches, there is a single unifying concept into which most of the most successful word classification systems can be transformed. This concept involves the quantification of the difference between two distributions — in this case, two distributions of word classes. Many successful word classification systems, to date, have worked by making operational definitions of the principles of structural linguistics. It remains to be seen, however, if these early successes can be improved upon sufficiently to make the structuralist approach any less unappealing to the mainstream of the linguistic community.

That these systems perform differently suggests that some measures are more appropriate than others; this highlights the need for a discriminating system evaluation tool and also suggests a useful line of research in mathematical approaches to language : why are some models better performers? what does this suggest about building better mathematical theories of language? The power of the bigram statistic is anomalous and perhaps even surprising from a traditional theoretical linguistic point of view. Church *et al.* [30] have made a significant start on an investigation of the differences between various statistics which are commonly used in computational linguistics and lexicography. They explain the difference in lexical use between the t-test, from traditional statistics, and mutual information, from information theory. With mutual information, they claim, it is difficult to test negative relations. This limitation with mutual information is discussed again in chapter 6.

In this chapter, we have seen that, with the advent of large natural language corpora and powerful computers, the statistical approach to language has been revitalised; in particular the relatively recent field of automatic word classification is beginning to produce encouraging results. In the following chapter we introduce a new automatic word classification system, and use it to classify letters, phonemes and words from several corpora.

Chapter 5

Exploring the Clustering Capacity of Mutual Information

5.1 Overview

This chapter describes some experiments in automatic word classification using unigram and bigram frequency statistics. This restriction of minimal word context is arbitrary — Brill *et al.*, for example have introduced the measure of generalised mutual information — yet, it still allows the researcher to investigate and exploit the structure inherent in linear sequences and to show the power of the concept of simple mutual information. Several of the most successful word classifiers use contiguous and non-contiguous bigram information simultaneously [53, 79, 143].

The following section explains the main algorithm in detail. Then the experiments are reported — each one investigates the performance of the algorithm at extracting different aspects of the structure of language, ranging from orthographically transcribed speech to phoneme sequences; from printed English to ancient Latin.

The last section describes the significance of the results; it concludes that simulated annealing based in mutual information is versatile enough to discover some structure in all of the main levels of language.

5.2 A Simulated Annealing Algorithm Based on Average Class Mutual Information

The representation introduced in section 3.6 — the structural tag — allows for words to be represented as s-bit binary patterns, giving access to s levels of word class information

simultaneously. This representation is similar to Schütze's *category space*, although the structural tag representation has the advantage that the information in each tag is immediately and independently available and can be used directly as a unique identifier of a word in a database of word-segment frequencies. Previously we have performed some experiments which use structural tags whose overall structure has been created by a top-down classification based on traditional linguistic categories (see figure 3.35). The following discussion explains how structural tags can be used in a locally optimal automatic word classification generator which delays making irreversible classifications within any level of the hierarchy and delivers a classification system which is informationally rich from the first classification level onwards.

The algorithm implements a type of simulated annealing which informally follows Brown's Law of Cumulative Complexity [159, 20] — this states that, during human cognitive development, if one structure contains everything that another structure contains and more, then it will be acquired later than the simpler structure. For us, this is equivalent to an algorithm which first treats all words in a corpus as if they belonged to only two classes — that is, the system contains one bit of class information about each word — and then attempts to find the optimal two-class word division. It then assumes that words belong to 2^2 classes and continues accordingly; the algorithm finishes when it reaches a classification with 2^s classes. The law seems to hold in first language acquisition and also in this and other algorithms (those of Wolff [159] and Elman [47]), because the more detail we have about a word or segment of words, the less frequently it occurs and hence the more time is needed to learn or identify the phenomenon. The algorithm can also be seen as a hypothesis generator and tester — the hypothesis being that the current classification contains the maximum amount of information about the structure of language, new hypotheses being generated by an exploration of nearby classifications in the current classification search space.

For a given vocabulary, V, a function f defines the relationship between any word and its corresponding unique structural tag. This function encapsulates an entire classification system. Many possible functions exist for given vocabularies; the task of automatic word classification is thus described in terms of searching the space of possible functions for that function which captures the relevant syntactic and semantic information. [1] Brown *et al.* have proven mathematically that any classification system whose average class mutual information is maximised will lead to class-based language models of lower perplexities.

[1] This function is used only as an interface between ASCII and TAG representations of words. Once we have made the mapping, not only does the TAG representation uniquely identify the word, but it also immediately contains all classifications of that word.

We recall equation 4.1, for computing the mutual information between two events :

$$I(x,y) = \log \frac{P(x,y)}{P(x)P(x)}$$

From this we can see that if the two events, x and y stand for the occurrence of certain word classes in a sample, then we can easily estimate the mutual information between two classes. In order to find the average class mutual information, we must first find all possible class bigram mutual information values and also how likely each of these pairs is. Summing the product of these values gives us an expression for the average class mutual information for a classification depth of s bits :

$$M_s(f) = \sum_{c_i,c_j} P(c_i,c_j) \times \log \frac{P(c_i,c_j)}{P(c_i)P(c_j)} \tag{5.1}$$

where c_i and c_j are word classes and $M_s(f)$ is the average class mutual information for a given classification f at bit depth s. It follows that the optimal classification, f^o can be found by computing

$$M_s(f^o) = \max_f \sum_{c_i,c_j} P(c_i,c_j) \times \log \frac{P(c_i,c_j)}{P(c_i)P(c_j)} \tag{5.2}$$

Unfortunately, no method exists at present which can find the optimal classification, but sub-optimal strategies exist which lead to interesting classifications. The sub-optimal strategy used in the current automatic word classification system involves selecting the *locally* optimal function between two functions which only differ in their classification of a single word. The starting function is generated by the computer's pseudo-random number generator. Its $M(f)$ value is calculated. Next, another function is needed which is very similar to the main function. It is created as a copy of the main function, with one word moved to a different place in the classification tree. Its $M(f)$ value is calculated. This second calculation is repeated for every word in V, we keep a record of the transformation which leads to the highest $M(f)$. At the end of a complete iteration through V, two scores are compared — the original $M(f)$ and the best of the new alternative $M(f)$ values. If the best alternative is better than the current function, the best alternative becomes the current function, and the process is repeated. By this method, words which at one time move to a new place in the classification hierarchy can at a later time, if the corresponding $M(f)$ value suggests it, move back. In practice, this does happen. Therefore, each transformation performed by the algorithm is not irreversible within a level, which should allow the algorithm to explore a larger space of possible word classifications.

The algorithm needs to be embedded in a system which works out the best classifications for *all* levels. This is done as follows. First, the highest classification level is processed. Since

the structural tag representation is implemented as a binary tree, this first level seeks to find the best distribution of words into two classes only. This is done by selecting best alternative functions and updating the main function until the best alternative is worse than the current one (see figure 5.1).

At this stage, it is concluded that, whatever later reclassifications occur among the words, at finer levels of granularity, they will always prefer to remain in the level 1 class to which they now belong. In other words, if many nouns now belong to class zero and many verbs to class one, subsequent sub-classifications will not influence the $M(f)$ score at classification level 1. This reasoning also applies to all lower classes (see figure 5.2).

Another way to describe this system is to focus on the individual word tags themselves. For every word tag of, say 16 bits, the algorithm looks at the most significant bit and answers the question : 'does this bit prefer to stay as it is, or does it prefer to flip?' This is why the structural tag tree is a binary tree. Whenever this question has been answered with the decision that the bit stays at its current value, for all words in V, then that level of classification is said to be complete. Notice that the next level of classification now does not have to evaluate the 4 possible combinations of 2 bits, since the first bit has been fixed. Thus, at each stage, only one bit's value per word is queried. One of the consequences of considering words and word classes as binary numbers is that classification is necessarily compressed into binary dimensions. At every level, there is a group of words which can only be separated in a binary way. This potential limitation is discussed further in the next chapter.

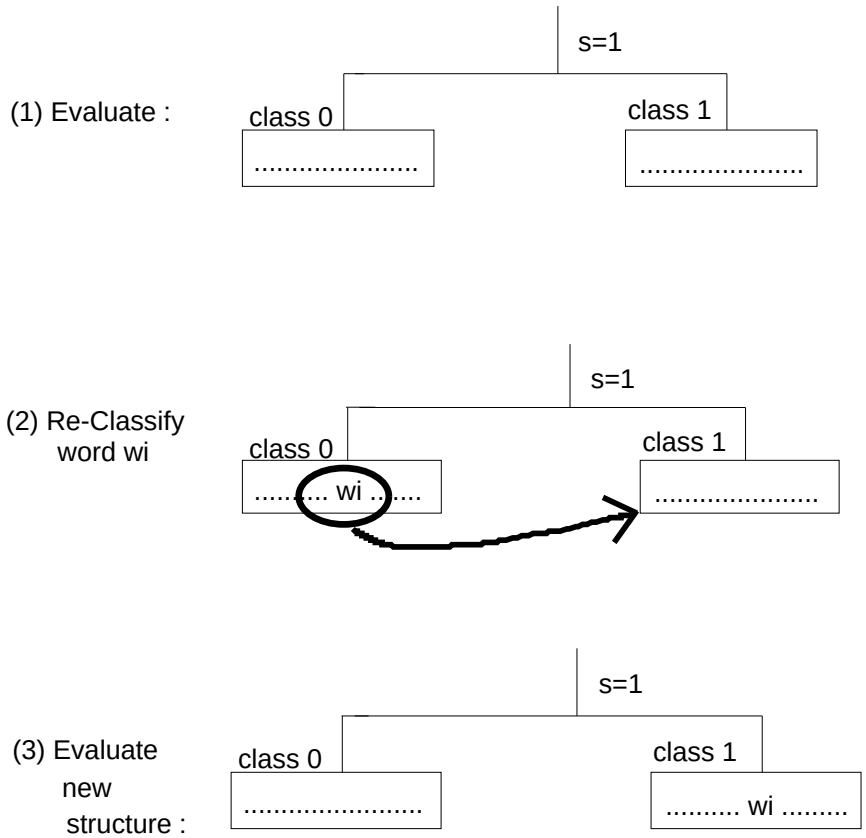
5.2.1 Worked Example

In order to describe more fully how the algorithm works, there follows a worked example of automatic clustering. Only four words are considered — the determiners ⟨`the`⟩ and ⟨`a`⟩ and the nouns ⟨`train`⟩ and ⟨`ticket`⟩. Maximum likelihood probability estimates based on the VODIS corpus are used to estimate all unigram and bigram probabilities. The example is concerned only with the first bit-level of classification and, in the initial classification, the words ⟨`the`⟩ and ⟨`train`⟩ have '0' as their first bit; ⟨`a`⟩ and ⟨`ticket`⟩ have '1' as their first bit.

Equation 5.1 is used to compute the average class mutual information for the one-bit classification :

$$M_1(f) = \sum_{c_i,c_j} P(c_i,c_j) \times \log \frac{P(c_i,c_j)}{P(c_i)P(c_j)}$$

We only concern ourselves with relative differences between classifications at any level, as opposed to the absolute average class mutual information values, so we can simplify our



(4) Move that word whose re-classification leads to the greatest increase in average class mutual information

(5) Repeat whole process until no word can be found whose move to another class leads to an increase in average class mutual information.

Figure 5.1: A default classification, of depth 1, is evaluated. Variations of this are evaluated, by moving one word in turn into its complementary class. The best variation is chosen to be the new default, whereupon the process is repeated until no variation is better than the default classification.

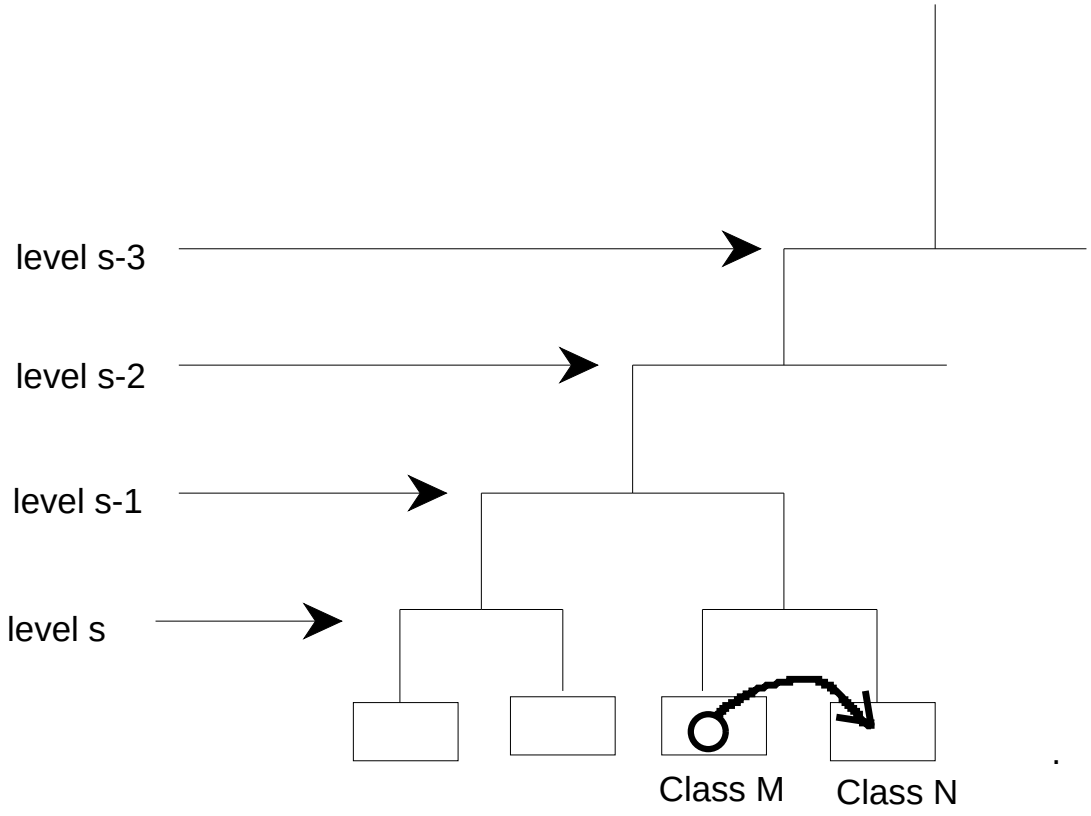


Figure 5.2: The algorithm is designed so that, at a given level s, words will have already been re-arranged at levels $s-1$, *etc.* to maximise the average class mutual information. Any alterations at level s will not bear on the classification achieved at $s-1$. Therefore, a word in class M may only move to class N to maximise the mutual information — any other class would violate a previous level's classification.

Word Flipped	MI value
(none)	-0.0635237
a	-0.0630931
the	-0.0599461
ticket	-0.062871
train	-0.0619043

Figure 5.3: Mutual Information when initial state is as follows : Class 0 = (train the) Class 1 = (ticket a); moving word ⟨the⟩ will lead to a better classification

calculations by removing the two shared constants of sum of unigram and bigram frequencies. We shall still refer to these mutual information analogue scores as mutual information scores for simplicity. Now, each word in turn has the first bit of its TAG representation flipped, and the resulting classification is evaluated by calculating its average class mutual information using the equation mentioned above. A word is permanently moved only when all words have been temporarily moved and evaluated and when the best temporary move — that is, the one which results in a classification with the highest mutual information — is better than the initial classification. In this example, the word ⟨the⟩ is chosen to be re-classified, leading to a new classification where only the word ⟨train⟩ has a zero bit and the other three words have a '1' bit. This new classification is processed again and this time the chosen word is ⟨ticket⟩ — its first bit is flipped. After this move, no single word can be moved to create a classification which is better than the current one — which has the two words ⟨train⟩ and ⟨ticket⟩ in one class and ⟨the⟩ and ⟨a⟩ in another. The fact that nouns have a zero-bit and determiners have a one-bit beginning is arbitrary, due to the initial random classification structure. The important distinction which the final classification captures is a relative one — this provides an example of Saussure's definition of *differance* [40]. Figure 5.3 shows the mutual information measure for the initial classification, together with new mutual information measures (log base 10) for each temporary flip. Figures 5.4 and 5.5 show the state of the classification when ⟨ticket⟩ is chosen to be moved, and finally, when no word can be moved with a corresponding increase in the mutual information measure.

5.2.2 Implementation Details

The main consideration in the design and implementation of the algorithm was computational efficiency. The algorithm uses a set of word unigram and word bigram frequencies as well as class unigram and class bigram frequencies. All data is read into memory, eliminating the

Word Flipped	MI value
(none)	-0.0599461
a	-0.0628638
the	-0.0635237
ticket	-0.0585254
train	-0.0636915

Figure 5.4: Mutual Information when state is : Class 0 = (train) Class 1 = (ticket a the); moving word ⟨`ticket`⟩ will lead to a better classification

Word Flipped	MI value
(none)	-0.0585254
a	-0.0619043
the	-0.062871
ticket	-0.0599461
train	-0.0630931

Figure 5.5: Mutual Information when state is : Class 0 = (train ticket) Class 1 = (a the); moving no word can lead to a better classification, so the processing at this bit level finishes.

need for any file access during the main part of the algorithm — see figure 5.6.

Word unigram and bigram frequencies will remain unchanged throughout the algorithm, so they are implemented as arrays : this allows faster access to the individual frequency values. Also, since, words are represented as TAGs, which are sixteen bit integers, a direct indexing system can be used for unigrams. That is, the word unigram frequency of the TAG whose value is `x` is accessed immediately by the expression `word-unigram-frequency-array[x]`. The same fast access data structure cannot be implemented with word bigrams, because there was not enough memory on the Sun^{TM} Workstation to deal with $V \times V$ arrays, for significant sizes of V. Instead, there is a $V \times 3$ array which consists of a linear sequence of word-pair and frequency values. Each iteration of the main algorithm needs access to the frequencies of all word-bigrams for a given TAG `t`, that is, for all `bigram(X,t)` and `bigram(t,X)` pairs. To this end, an array of linked lists is implemented — the array index identifies the TAG `t` which is being processed during the current iteration and the linked list contains a list of indices to the word bigram and frequency arrays. This is computationally less expensive — whenever a word is being temporarily re-classified, its TAG value acts as a direct index into an array of linked lists. The linked list which the TAG index points to is itself a list of indices into

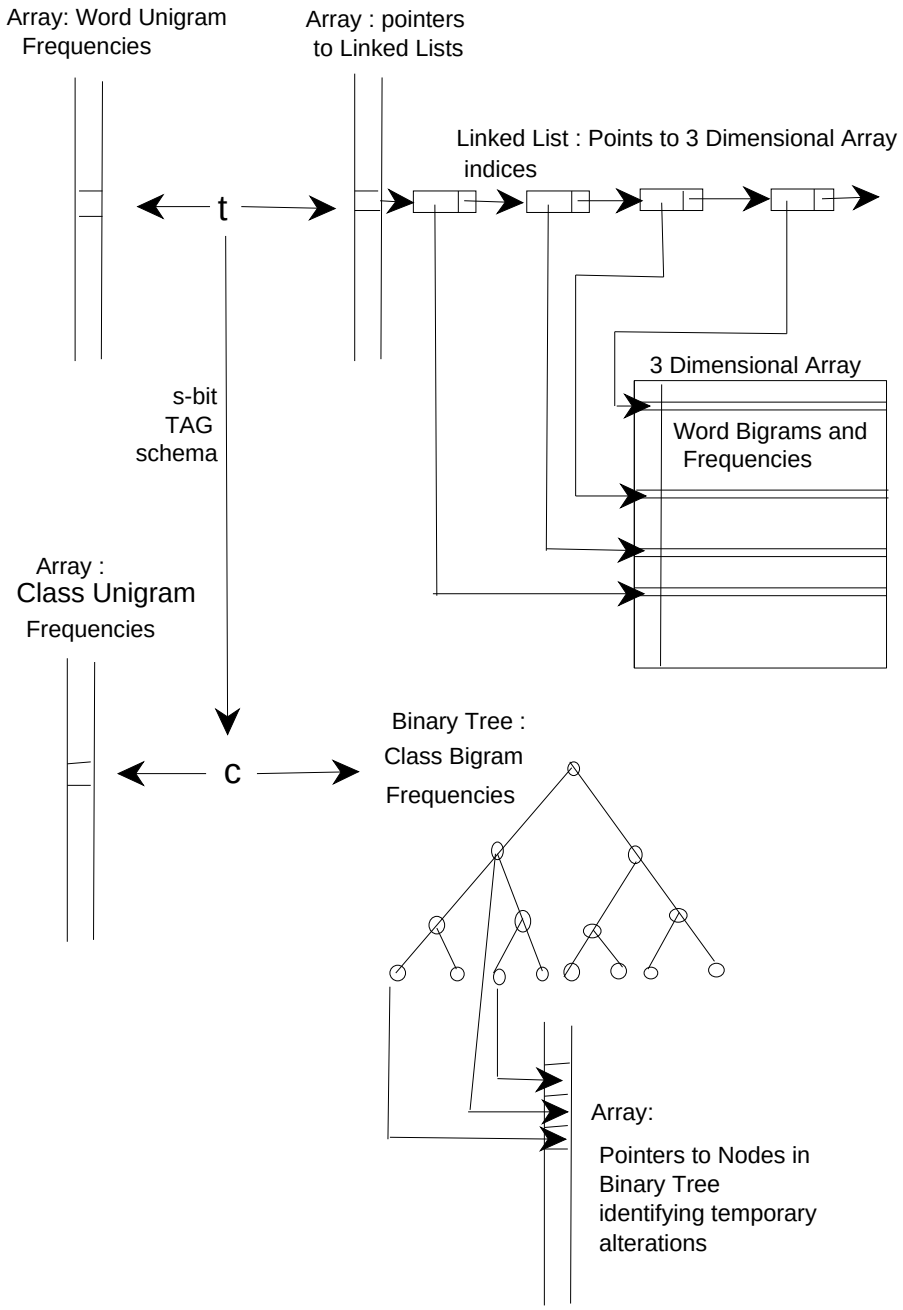


Figure 5.6: Main data structures of the annealing algorithm. The points of entry are with word TAG t and the class c, of that tag. TAG t acts as an index into a word unigram frequency array; it also indexes a linked list, containing a set of index references to all of the word bigrams which contain TAG t. Similarly with class c, except that class bigrams are stored as a binary tree. An extra array of pointers to nodes in this tree allows fairly efficient re-setting of the class database when needed.

another array containing all of the relevant bigrams which include the original word; the list also contains the relevant bigram frequencies.

When implementing the unigram class database, arrays can be used in the way described above, but the bigram class database should not use arrays because each iteration will alter the database and arrays cannot be altered efficiently — a binary search tree is implemented to make identification of class pairs fairly efficient; also, an array of pointers to various nodes in this binary tree allows the tree to be reset quickly, without having to to do superfluous searching.

Analysis of running time shows that the main part of the algorithm is $O(n^3)$, making usual assumptions about an ideal computational system [156]. To give results in a reasonable time, only the most frequent words of a vocabulary are given as input to the algorithm. Zipf's law, however [164], ensures that this accounts for the majority of the tokens of a corpus. The law states that there is a constant relationship between the rank of a word in a frequency list and the frequency with which it is used in a text. Although the law does not hold for words of the very lowest and very highest frequencies, the law has empirical support for the vast majority of word frequencies [119].

Dunning [44] points out that rare words are not distributed Normally, so the technique may have problems when being applied to these word types (section 6.5 contains a fuller discussion of the implications of Dunning's claims and section 7.5 introduces another clustering method which can be used with medium-frequency words).

5.3 Repeating Elman's Experiment

5.3.1 Elman's Original Results

This section re-describes the salient details of one of the experiments performed by Elman in [45]. The grammar which generates the language upon which his experiment is based is, according to the Chomsky classification, type 4 (regular, or finite-state). Its production rules are shown in figure 5.7. Many of the words belong to two or more word classes. The sentence frames encode a simple semantics — noun types of certain classes engage in behaviour which is unique to that class. Elman generates a 10,000 sentence corpus to be used as the training corpus. Each sentence frame is just as likely to be selected as any other; similarly, each word member of a particular word group has an equiprobable selection chance. No punctuation is included in the corpus; so sentence endings are only implicitly represented — for example, the segment stream ⟨`cat smell cookie dog exist boy smash plate`⟩ contains a three word

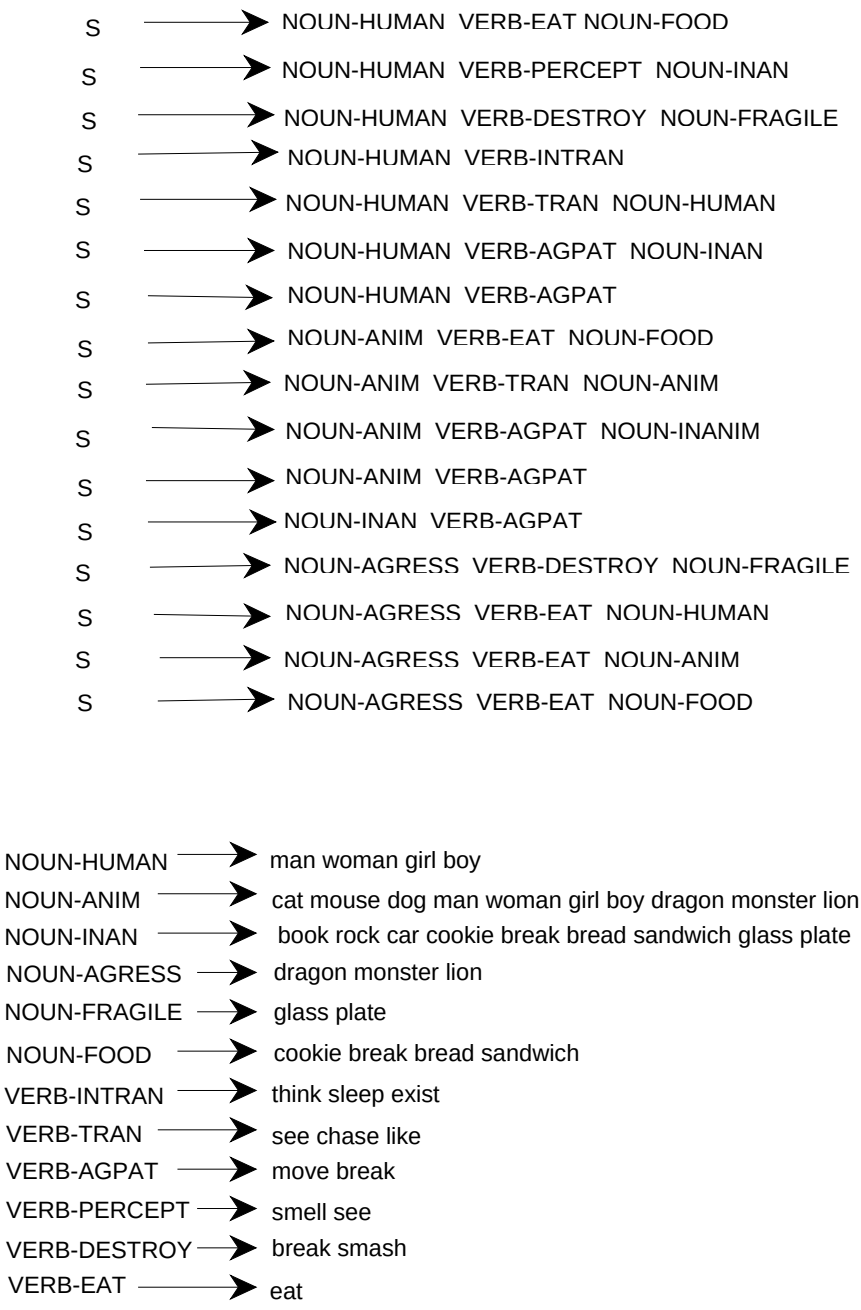


Figure 5.7: Elman Grammar. There are sixteen non-terminal rules and twelve terminals. Notice also that terminals can belong to more than one word class — for example, ⟨`break`⟩ is an inanimate noun, a food noun, an agent-patient verb and a destroy verb.

sentence followed by a two word sentence followed by another three word sentence, but any algorithm operating on this input will have to induce sentence boundaries.

The task can be summarised as follows : given a linear sequence of continuous sentences, the system must identify that words like ⟨`see`⟩ and ⟨`smell`⟩ are similar, and that words like ⟨`lion`⟩ and ⟨`monster`⟩ are similar. The difficulty of this task becomes greater when the only statistics being used are unigram and bigram word frequencies.

The Elman Net Prediction Performance

After training, Elman's net was tested on an unseen set, generated by the same underlying grammar. The network's performance was poor — only achieving a prediction performance error rate slightly above chance.Elman then presented the training data to the net a further four times, but the prediction was still poor. He claims that, with even more training, the net could have improved its performance further. But this was not the main goal of the experiment; instead, hierarchical cluster analysis was performed on the averaged hidden unit activations for each of the 29 words.

Elman's Similarity Tree

Figure 5.8 reproduces the similarity tree which cluster analysis of the recurrent net produced. His analysis reveals that the network has learned most of the major syntactic differences and many of the semantic ones which are coded in the original language. For example, there is a clear distinction between verbs and nouns; within the main class of nouns, there is a clear animate-inanimate distinction; within that, the classes of agent-patient, aggressor and non-human animal have been induced. This important result shows that computational linguistic systems can operate using self-organised distributed representations of the higher level hierarchies of grammar and semantics. The analysis is not perfect : the most important distinction is considered to be between a handful of inanimate noun objects (bread, cookie, sandwich, glass and plate) and the rest of the vocabulary. In other words, the analysis which this system reveals is not ideal for many-level class-based statistical language modelling since many of the early class divisions do not capture coherent language structure.

5.3.2 Mutual Information Clustering on the Elman Corpus

This section describes the results obtained when the mutual information based algorithm described earlier is applied to a similar test corpus. Elman's grammar of figure 5.7 was used to produce a corpus of 10,000 sentences with no sentence breaks. Unigram and bigram

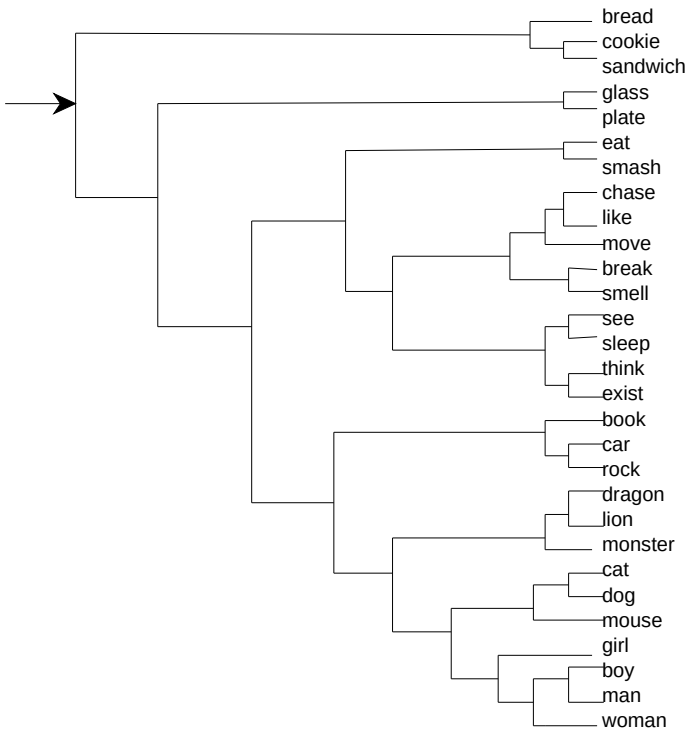


Figure 5.8: Elman's results — A Cluster analysis of the hidden units of a trained recurrent net, showing the major verb-noun distinction, as well as many other syntactic and semantic fine-grained distinctions.

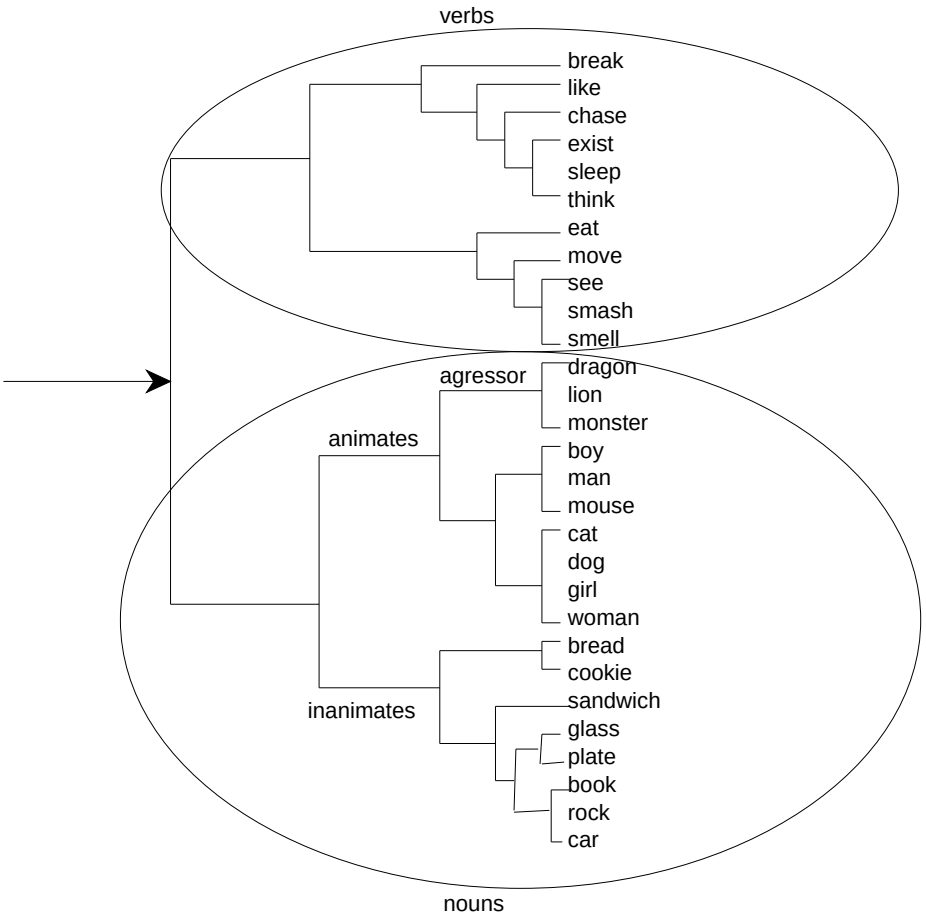


Figure 5.9: Class hierarchy using structural tags and average class mutual information for the Elman experiment. Sub-classifications are only displayed up to the first point where they are misclassified or when they correctly identify a class.

word frequency statistics were generated. The structural tag word classification algorithm described above was implemented and applied to the initial function which randomly assigned tag values to the 29 words. Figure 5.9 shows the important classification decisions made by this algorithm. This explicit hierarchy could now be inserted in a statistical language model based on the principle of defocusing outlined in figure 3.37. Also, unlike the Elman classification (see figure 5.8), informationally useful class structure exists from level 1 onwards. This algorithm also produces a classification some features of which are qualitatively better than Elman's — all nouns and all verbs are separated (see chapter 6.2 for a discussion of the problems involved with comparing classifications); all animates and inanimates are separated. The multi-context noun/verb ⟨`break`⟩ is identified as different from other verbs; intransitive verbs cluster together and the aggressive nouns are identified. This algorithm doesn't recapture the complete syntax and semantics of the language — human nouns and

non-aggressive animate nouns remain mixed and the food noun cluster failed to attract the word ⟨`sandwich`⟩.

The reasons for the weaknesses in the fine details of the classification spring from the sub-optimal strategy of the current algorithm, combined with the initial word classification state. This experiment was repeated several times, each time resulting in a classification whose overall structure was similar but whose fine detail was slightly different. One run, for example correctly differentiated between small animals and humans, but failed to recognise food nouns as a complete group. Another run identified food nouns perfectly but failed to separate aggressors from other animates. It is postulated that the optimal function would correspond more fully to the syntactic and semantic distinctions which generated the language. Another limitation involved the relative sparseness of the data — this is supported by the fact that when level 1 classes were being evaluated, *i.e.* when class frequencies were very high, then the main distinctions were learned. Whenever more fine-grained classifications were involved, the corresponding class frequencies were smaller and so the probability estimates were less reliable.

5.4 Clustering on a Small Spoken Language Corpus

The mutual information results from the Elman language are successful, especially at broader levels, partly for the following reason : most of the dependencies are caught *directly* by bigram relations; those that are not are caught indirectly by the shortest of all possible transitive chains — $\langle AB \rangle$ $\langle BC \rangle$. In natural language, this chain is usually much longer : involving features like verb agreement, pronoun reference and higher level semantic and pragmatic features like selectional constraints. Languages which include recursive rules of generation can, in theory, generate infinitely long sentences. Natural language utterances demonstrate a limited type of embedding — the deeper the degree of embedding, the more difficulty native speakers have in processing such constructions.

Differences between Spoken and Written Corpora

The VODIS corpus provides a useful starting point for an investigation into the problems which arise when word classification is attempted using natural languages. O'Boyle [119] has shown that VODIS is inherently a *low entropy* corpus. That is to say, the grammatical structures and vocabulary range which are found in the corpus are heavily constrained — just as a sociolinguistic investigation of the sorts of linguistic behaviour which occur in these

sorts of exchange over telephones should show a similarly well-circumscribed pattern. This effect is a well-observed phenomenon [82]. Stephens and Beattie [151] claim that telephone travel query conversations are highly structured and that a significant part of this structure resides in prosody; even syntactic and semantic cues for guessing which word comes next are modulated by prosodic effects. Derouault and Merialdo([42] suggest that natural language utterances are more complex structurally than written ones; if this is the case, then the low entropy of VODIS must be related to its domain-specificity. This is a conclusion shared by Jelinek *et al.* [86]. While this makes the task of next word prediction relatively easier, it also makes consistent word classification more difficult. Fisher and Riloff [55] also assert that domain specific corpora allow for semantic regularity to emerge if the corpus size is small and the information extracting statistic is the t-test. The question of the inherent complexity of spoken (transcribed) versus written corpora is not uncontroversial. Smith [146] recognises that, in terms of Saussurian linguistics, phonetic and orthographic signifiers are equally valid. He adds, however, that orthographic changes tend to be able to be regulated by political and cultural control easier than phonetic changes. Also, orthographic descriptions of language have some limited devices for simulating prosody — punctuation, use of different typefaces and, in some domains (*e.g.* USENET groups) iconic facial representations. Smith concludes that these extra channels of communication tend to be so much more limited that written language utterances make up for the deficit by exhibiting a slightly higher degree of clarity — they contain less noise and other performance related errors. Smith also demonstrates some of the differences between written and spoken corpora : there is more regularity in written languages in the morphology of lexical verbs; the written plural of 'dog', 'cat' and 'horse' involves adding '-s', whereas the phonetic pluralisation involves adding /s/, /z/ and /es/ respectively. Similarly, the past tense of 'shove' and 'talk' involves adding '-(e)d' while the phonetic equivalent involves adding /d/ and /t/ respectively. Some related words are obviously related orthographically, but less obviously so phonetically (*e.g.* 'real/reality' and 'human/humanity'). Finally, Smith points out that the phonetic /for/ can map to 'for', 'four' or 'fore'. He accepts that reverse effects exist, but maintains that written texts are less complicated than speech-transcribed texts which have lost prosodic information.

Ideally a desirable classification system should map the word ⟨british⟩ into the space of adjectives, and further into the space of adjectives which modify only certain types of noun. This will occur if the corpus contains utterances like ⟨british man⟩, ⟨british empire⟩, ⟨british rail⟩, ⟨british airways⟩, ⟨british nationalist⟩, ⟨british telecom⟩, *etc.* If, however, the corpus has only one substantial context for the word ⟨british⟩ to precede —

e.g. ⟨`british rail`⟩, as in the VODIS corpus — then it is not going to be mapped to a homogeneous space of adjectives. Also, if the verb ⟨`help`⟩ is to be mapped near root forms of lexical verbs, then a corpus which contained at least ⟨`i help`⟩, ⟨`you help`⟩, ⟨`they help`⟩, ⟨`we help`⟩ would be desirable. Again, if certain key contexts dominate, this too will skew the classification structure. The less representative the set of contexts of a corpus — that is, the more domain specific that corpus is — the less likely a traditional linguistic classification will be generated.

It should be noted that the classification scheme derived by a perfect word-classification system, maximising average class mutual information, will be the best classification *for that corpus*; if this classification should happen to correspond only approximately to a traditional linguistic conception, then the disparity lies with the corpus, which doesn't provide sufficient contexts for the classification scheme to work.

In practical terms, this implies that corpora with entropies closer to that of the entropy of the underlying language should be used to generate linguistically appealing classes. It is still useful to try experiments on the low-entropy VODIS corpus, since the classification should suit the corpus, and secondly, since at least some patterns should be seen from the initial levels of classification. Another practical consideration is the fact that the algorithm has very high computational overheads; also, words which occur with only moderate frequency tend to lead to unreliable context estimates. For the purposes of comparison, it is noted that the training set used by Brown *et al.* in their class-based research is 6330 times larger than the VODIS training set used in the following experiment. Consequently, the classes derived from the smaller corpus should be correspondingly less impressive.

An experiment was performed using the automatic word classification system and the top three hundred frequently occuring words from the VODIS corpus. Each evaluation of $M(f)$ involved only unigrams and bigrams of these frequently occuring words. Brill *et. al.*, Brown *et. al.* and Schütze all perform their automatic word classification experiments on only the most frequently occurring words, since low frequency unigram and bigram counts are less reliable. Dunning [44] and Fisher *et al.* [55] discuss the problem of statistical reliability in more detail.

Analysing the Results

Classification levels will be examined in order to discover the distribution of some syntactic and semantic classes. Balanced distributions suggest that a classification has not clustered any significant information; uneven distributions suggest the opposite. This is because clas-

sifications which are random tend to distribute words evenly. The uneven nature of word frequency statistics suggests that useful classifications should also be uneven; also, word classes are uneven : closed classes have small, unchanging memberships, while various open classes have varying membership numbers.

A binary tree was drawn upon completion of the automatic classification of the fifth level — generating 32 classes at this level. Figure 5.10 shows the initial random distribution of approximately the most frequent three hundred words in a VODIS corpus which has been formatted to remove all capitals, most punctuation and which replaces all number instances with the corresponding word. [2] The words are evenly distributed and no pattern can be discerned.

Figure 5.11 shows the state of the classification after five levels of self-organisation have been completed. Some major syntactic and semantic distinctions have been discovered by the algorithm. At the first level, there seems to be a broad distinction between content words and function words. Most determiners are mapped into a close region of the classification space, as are prepositions, low order cardinal numbers, place names, and modal verbs. Some classes at level five, especially content classes, are large and contain many verbs and nouns. At deeper levels, these separate out in the expected ways.

Content words are less frequent than function words in corpora. Even though function words have high relative frequencies in any corpus, these frequencies may still be unreliable if the corpus is too small. In order to discover accurately the taxonomic relations between many of the function words, large corpora need to be used. The situation might seem worse for content words, but the problems are alleviated if the corpus in question is highly domain-specific. Low information function words usually occur in just the same types of context whether the corpus is domain-specific or not. With high information content words, the more domain-specific the corpus is, the greater the likelihood that those words are used in only limited contexts. With the VODIS corpus, the domain involves train times, ticket prices and travel locations. The first two imply utterances using numbers and the last implies utterances using place names. The contexts involving these domain-specific content words should be more limited and hence more statistically significant than would otherwise be the case. Figure 5.12 shows the distribution of numbers throughout the tree. There is an obvious clustering effect. One reason why the numbers should have two distinct loci within the hierarchy is because numbers appear as amounts and also as times. The number segments

[2] For example, '10.30' becomes 'ten thirty', '12.05' becomes 'twelve oh five' and '$9.99' becomes 'nine pounds ninety nine'

Figure 5.10: VODIS Vocabulary : Initial random binary tree, to level five, with approximately the most frequent three hundred words distributed randomly throughout the classification hierarchy.

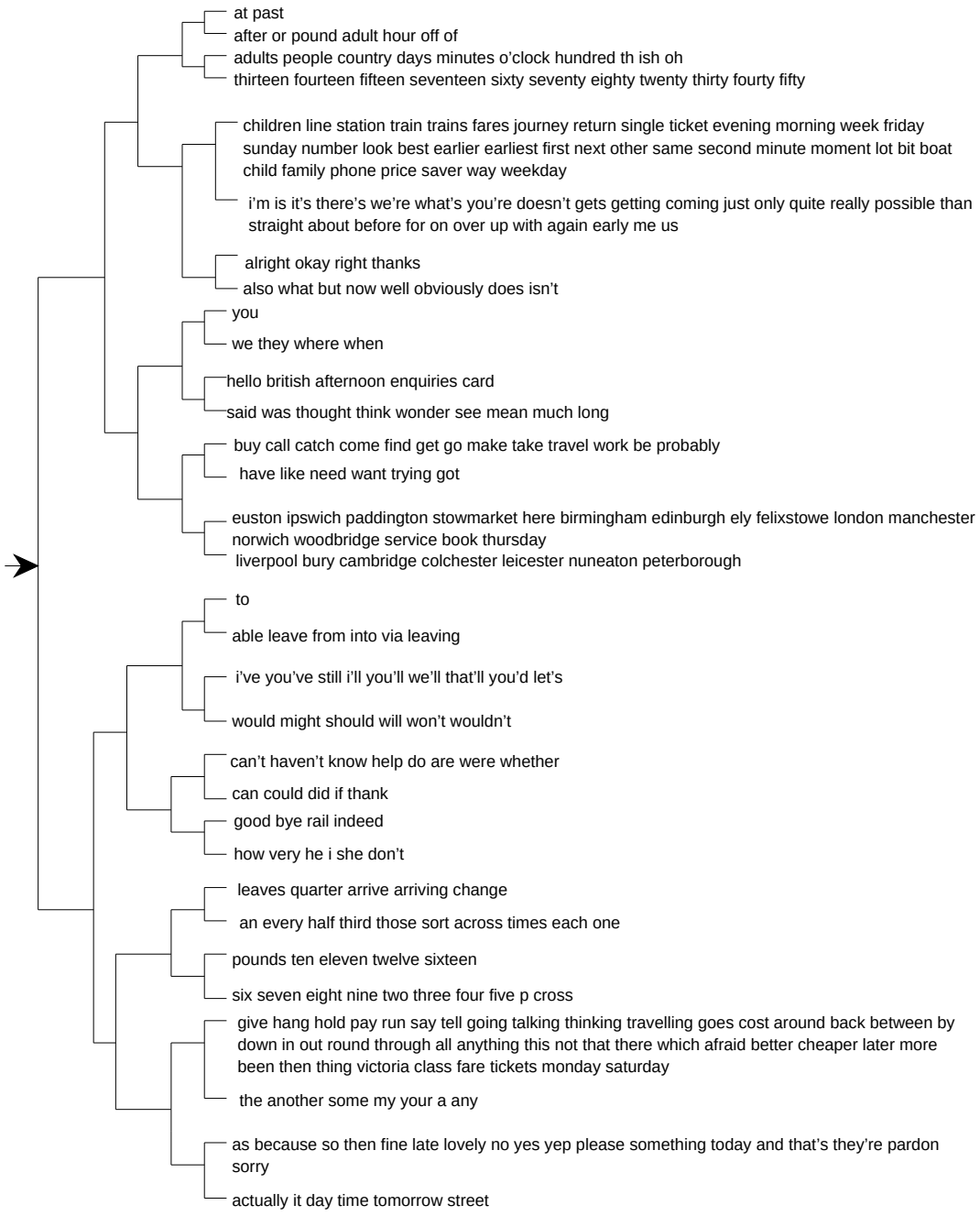


Figure 5.11: State of the self-organised VODIS classification after five levels. Major syntactic and semantic distinctions become clear.

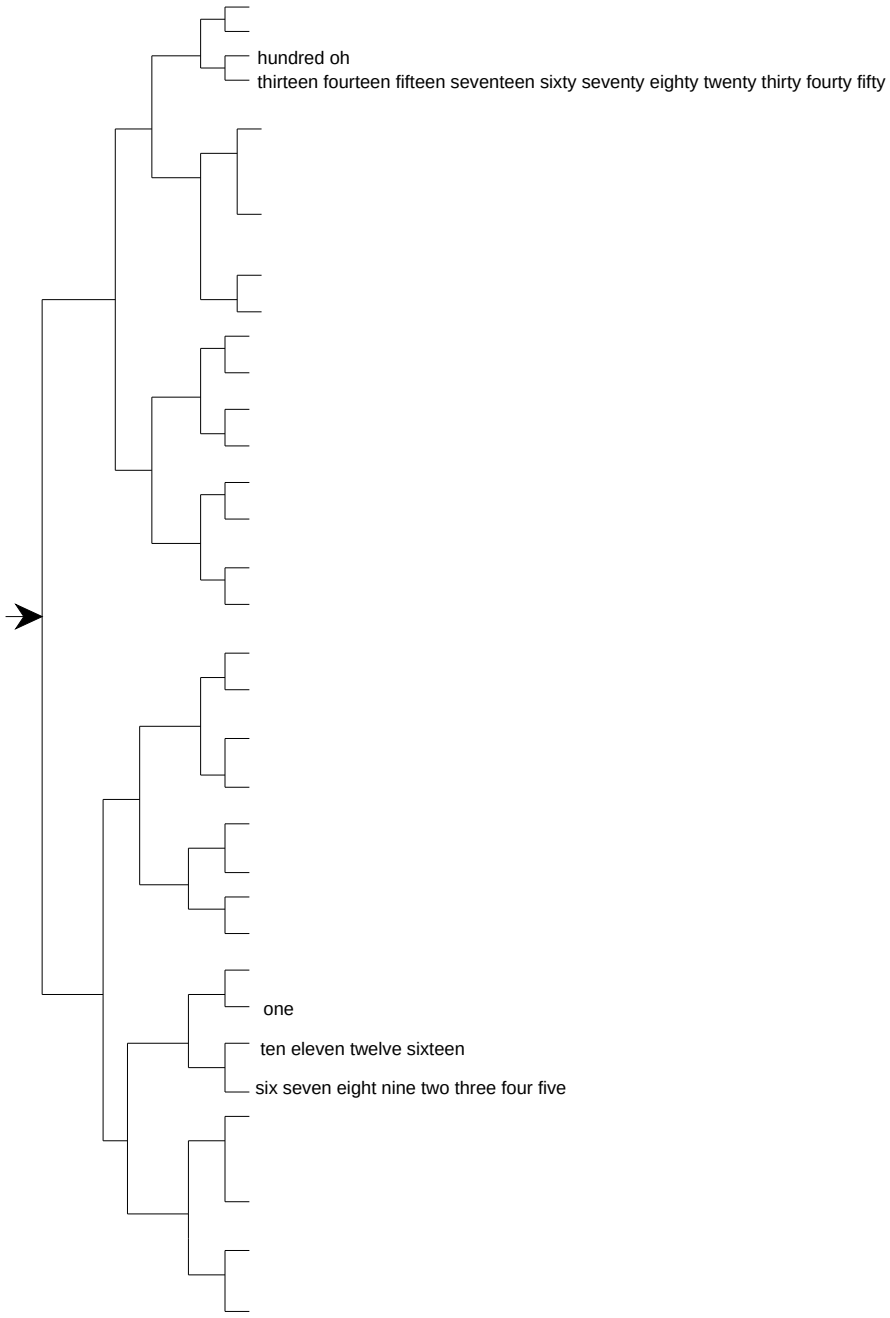


Figure 5.12: VODIS Tree : Two main regions where numbers are clustered. In one, the numbers ⟨`one`⟩ to ⟨`twelve`⟩, plus ⟨`sixteen`⟩ are clustered; in the other, multiples of ten plus some numbers from ⟨`thirteen`⟩ to ⟨`seventeen`⟩.

⟨one⟩ to ⟨twelve⟩ can be amounts, but can also be hour times. This is not true of the numbers in the other cluster. Also, whether numbers are used as amounts or as times, numbers usually come in groups : for example, the time segments ⟨two thirty⟩, ⟨nine oh five⟩ and ⟨eleven fourty five⟩; and the amount segments ⟨ten fifty⟩ and ⟨thirteen seventy⟩. Average train fares are usually less than twenty pounds, so this could explain why all of the higher multiples of ten appear together in their group. This group is situated with noun-like words because they have similar distributional properties to nouns. The smaller numbers — ⟨one⟩ to ⟨twelve⟩ — have modifier-like behaviour since they usually come just before the noun-like numbers, or they modify the words ⟨pound⟩, ⟨pounds⟩ and ⟨o'clock⟩. Figure 5.13 isolates the cluster positioning of place names. The majority get mapped into a very close part of the tag space. The domain-specificity of the original corpus becomes apparent here — the fact that the corpus was constructed from Ipswich British Rail telephone enquiries is reflected in the destinations which callers enquire about. Also, the word ⟨here⟩ is mapped into this space because in most of its contexts, ⟨here⟩ refers to Ipswich.

Figure 5.11 contains many more interesting clusters, including the following groupings which occurred at classification level five : ⟨alright⟩, ⟨okay⟩, ⟨right⟩, ⟨thanks⟩ — signs of understanding or agreement; ⟨would⟩, ⟨might⟩, ⟨should⟩, ⟨will⟩, ⟨won't⟩, ⟨wouldn't⟩ — modal verbs, which are also close in space to the group : ⟨i've⟩, ⟨you've⟩, ⟨still⟩, ⟨you'll⟩, ⟨we'll⟩, ⟨that'll⟩, ⟨you'd⟩, ⟨let's⟩. Also the group : ⟨an⟩, ⟨every⟩, ⟨half⟩, ⟨those⟩, ⟨sort⟩, ⟨across⟩, ⟨times⟩, ⟨each⟩, ⟨one⟩ — occurs close to the modifier-like numbers; also, nearby is the group : ⟨the⟩, ⟨another⟩, ⟨some⟩, ⟨my⟩, ⟨your⟩, ⟨a⟩, ⟨any⟩. The words : ⟨as⟩, ⟨because⟩, ⟨so⟩, ⟨then⟩, ⟨and⟩ — get mapped into the same group, as do the words : ⟨no⟩, ⟨yes⟩, ⟨yep⟩, ⟨fine⟩, ⟨lovely⟩, ⟨please⟩, ⟨pardon⟩, ⟨sorry⟩.

5.5 Clustering on a Medium-sized Written Language Corpus

The classification algorithm was applied to the LOB (Lancaster-Oslo/Bergen [87]) corpus of one million words, taken from a variety of domains. The corpus is tagged, though the tags were removed for this experiment. This new source of language examples was used to alleviate two problems. First, the corpus is less domain specific, and so hopefully a more balanced set of contexts exists for any given word; second, the corpus is larger, increasing the statistical significance of the contexts. Again, only the most frequent words were considered.

Figure 5.14 shows the state of the classification tree before the automatic classification algorithm is executed. Words are distributed randomly throughout the structure. Figure 5.15 shows the final state of the hierarchy for the first five levels. There are many important

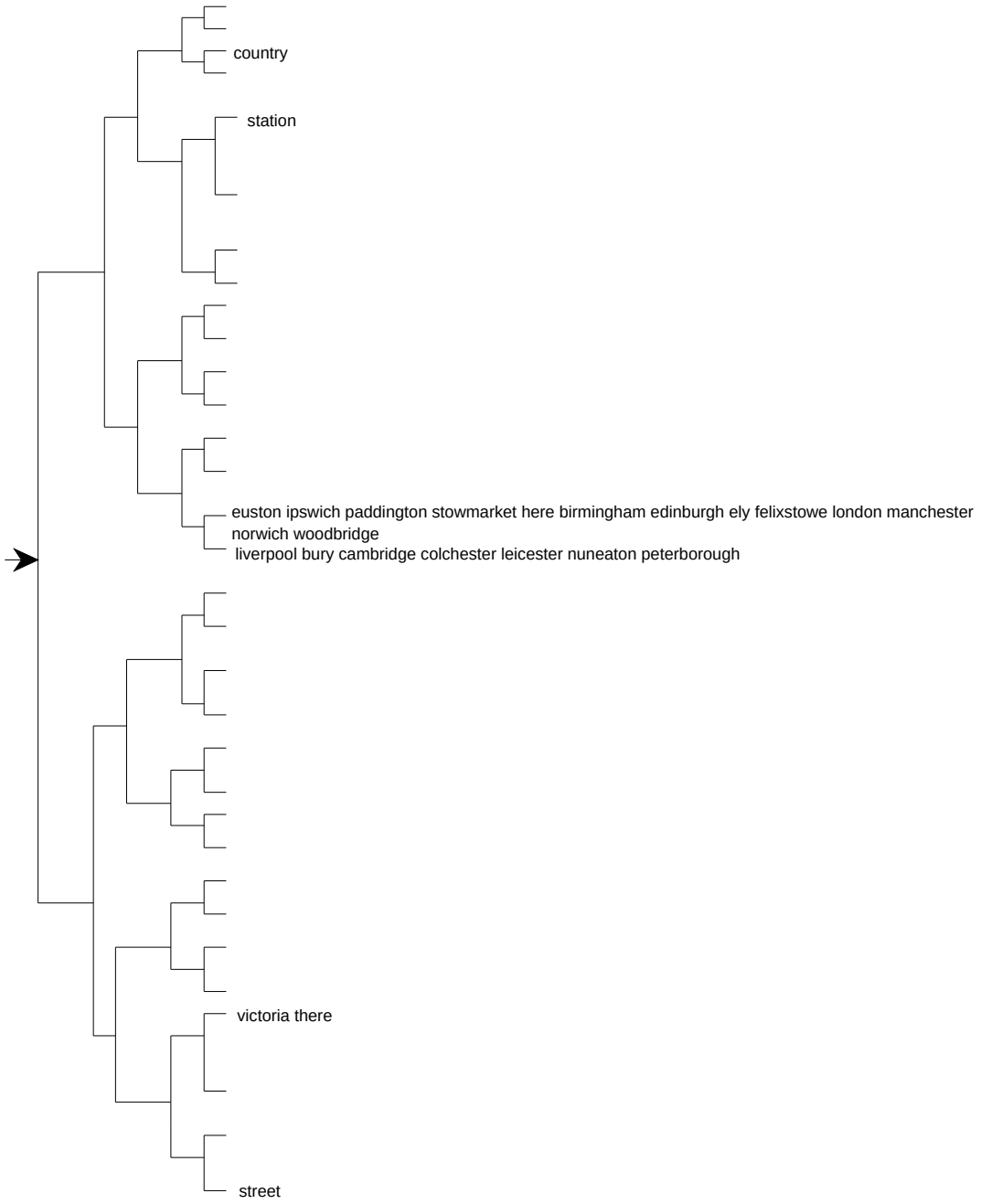


Figure 5.13: VODIS place names demonstrate a high degree of clustering in a small region of the tag space. The corpus from which this classification was derived was collected from incoming calls to Ipswich Town Rail Station; notice how the words ⟨`ipswich`⟩ and ⟨`here`⟩ get mapped into nearby points in the space.

syntactic relations which have been discovered by the algorithm. These can be described more clearly by looking in more detail at some of the more interesting arrangements of words.

Results — Finer Classification Detail

Examples of finer classification detail are given in figures 5.16 to 5.24. The first of these, figure 5.16 shows a complete description of that area of the classification hierarchy which starts with the five bits 00010.

This region contains pronoun-like words. [3] The figure shows that ⟨you⟩, ⟨there⟩ and ⟨it⟩ form one sub-cluster while ⟨I⟩, ⟨we⟩, ⟨he⟩, ⟨she⟩, ⟨they⟩ and ⟨who⟩ form another. One reason for this division may be because the words from the former cluster occur in sentences such as :

```
it is raining
there is no need for that
if you want to succeed in life then work hard
```

That is, in sentences where the pronoun doesn't necessarily refer to a particular person. Another feature of this pronoun classification is closeness of clear plurals — ⟨we⟩ and ⟨they⟩ — and the closeness of the pronouns ⟨he⟩ and ⟨she⟩. Figure 5.17 shows some of the detail of the region of space identified by the bit pattern 0000. The majority of these words share some determiner-like behaviour. For example, the titles ⟨Dr⟩, ⟨Sir⟩, ⟨Miss⟩ and ⟨Mrs⟩ are usually found before nouns which are proper names. Numbers can also appear before nouns, as can other quantifying determiner words like ⟨another⟩, ⟨each⟩, ⟨some⟩, ⟨many⟩, ⟨several⟩, ⟨all⟩ and ⟨both⟩. Possessive determiners are represented here and also cluster — ⟨our⟩, ⟨its⟩, ⟨their⟩, ⟨his⟩, ⟨my⟩, ⟨your⟩ and even ⟨whose⟩. The word ⟨no⟩ appears in this region, implying that segments like

```
There was no reason to lie
```

represent more significant contexts for the word than segments like

```
did you lie ?
no
```

The next figure — 5.18 — shows more detail in the part of the tree identified by the bit pattern 00011. Symbols like ⟨*'⟩ are conjunction-like in the sense that they mark off distinct

[3] Section 6.2 describes the difference in distribution between +nominative pronouns, seen in figure 5.16, and -nominative pronouns, seen in figure 5.20.

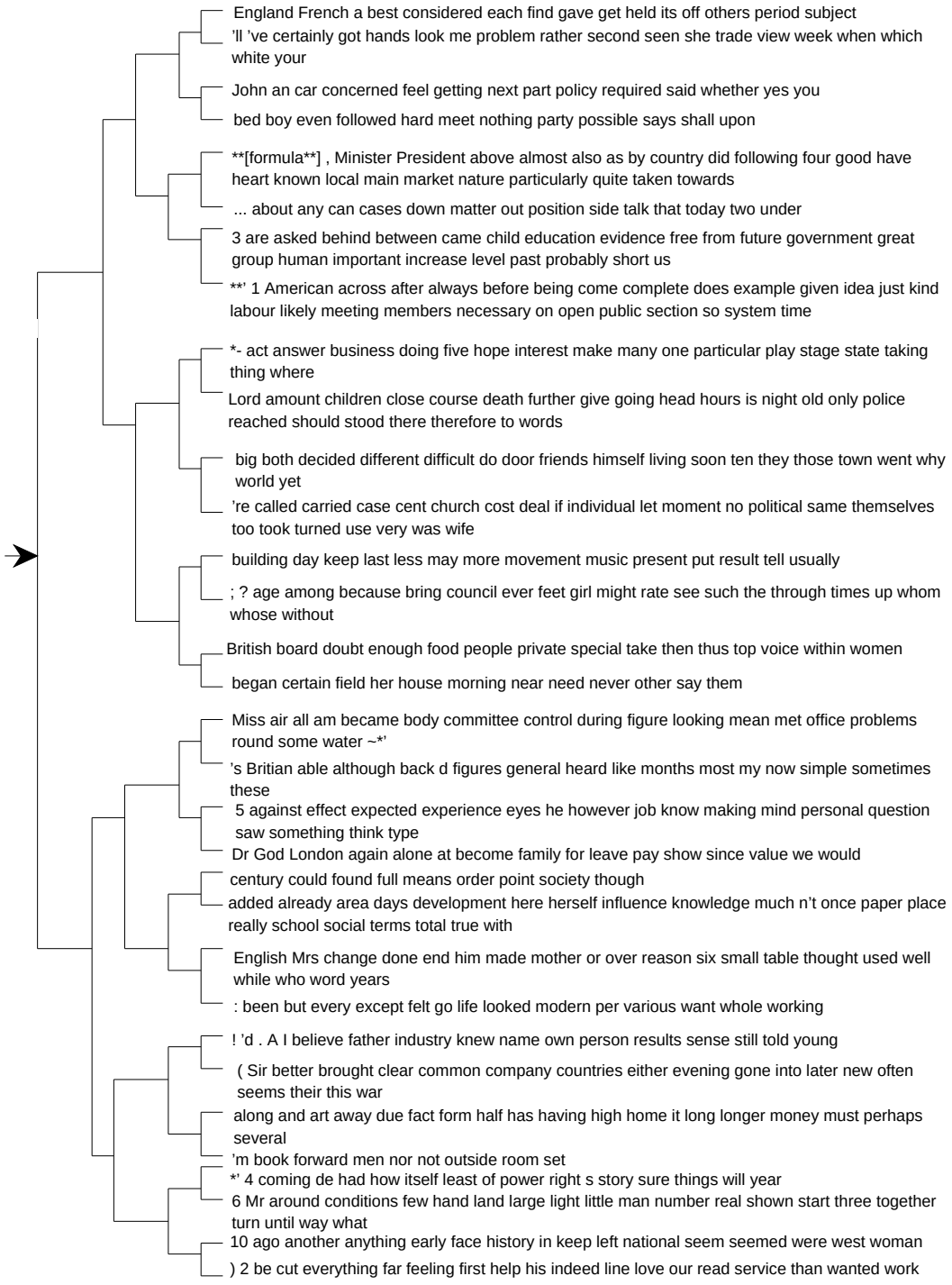


Figure 5.14: Initial random distribution of the most frequent words from the LOB corpus. No discernable pattern exists in the classification. Only the first five levels of classification are given here.

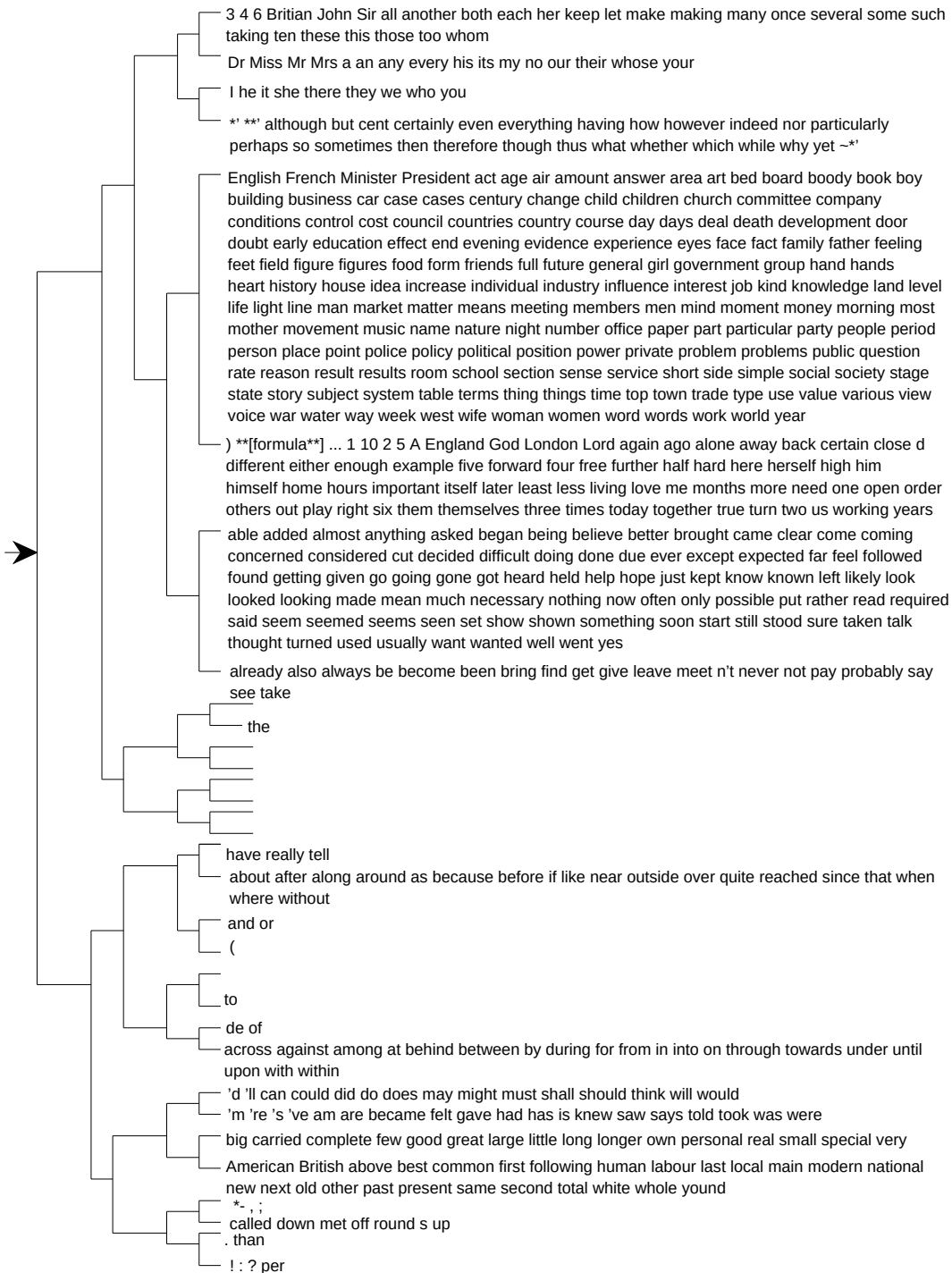


Figure 5.15: Final distribution of the most frequent words from the LOB corpus. Only the first five levels of classification are given here, but important syntactic relations are discovered.

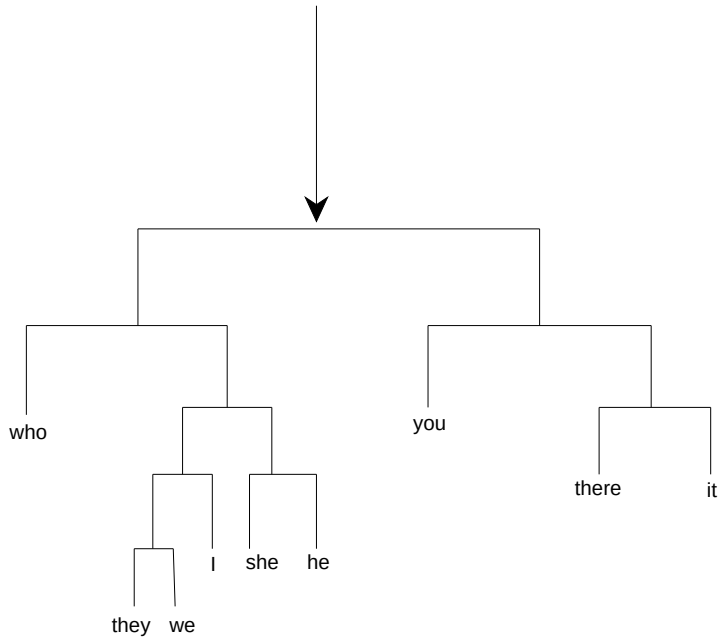


Figure 5.16: Detail of relationship between words whose final tag value starts with the five bits 00010.

but possibly related propositions without having occurred as frequently as more traditional punctuation symbols. Many of the linking conjunctions in this figure are subordinating conjunctions, as opposed to the more neutral co-ordinating conjunctions such as ⟨`and`⟩ and ⟨`or`⟩, which appear in a different region.

Figure 5.19 shows some more of the detail of the set of words whose final bit pattern begins with 0011. Many of these words have noun-like properties; a significant minority have noun-modifier-like properties. The bit pattern 0011, therefore, has done much of the work in identifying the syntactic features of these words, yet sub-classification continues in a meaningful way. In an ideal classification, a broad distinction should exist between singular and plural nouns; this is not seen. Again, the reason is due to the small size of the corpus, which leads to an insubstantial sample of singular-plural differentiating contexts. There is, however, a qualitatively significant difference between words in the 00110 and the 00111 classes with respect to singular and plural words. There are fewer plurals than singulars overall, so it is easier to give the plural distributions. For class 00110, there are two definite plural nouns, while in class 00111, there are seventeen. Even within class 00111, an uneven singular-plural division can be described. The figure does demonstrate some semantic similarities between words. Most apparent are the classes containing the singular human types
:

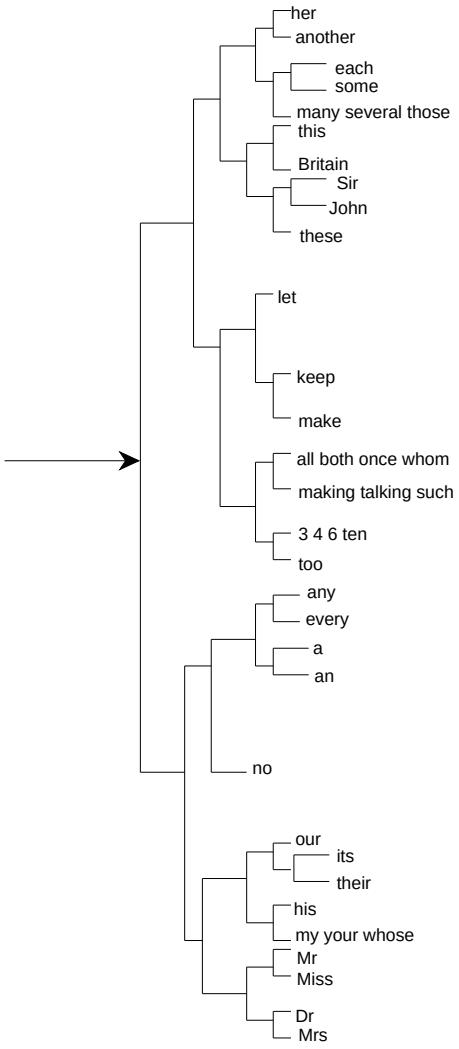


Figure 5.17: Detail of relationship between words whose final tag value starts with the four bits 0000. Many of these words exhibit determiner-like behaviour.

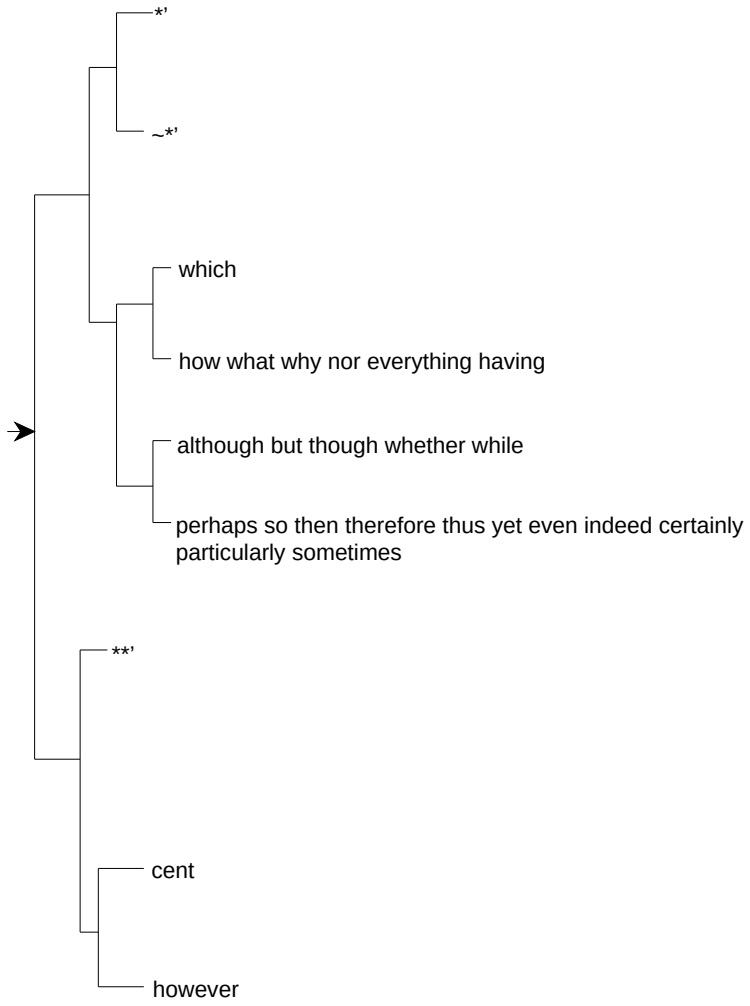


Figure 5.18: Detail of relationship between words whose final tag value starts with the five bits 00011. Many of these words exhibit conjunction-like behaviour.

boy girl child man woman person

the singular time measures :

day night morning evening week year moment

the body parts (plus some family member nouns) :

eyes face feet hand hands head heart mind voice father mother wife

and a related group of plural nouns :

children friends people men women countries words things

Other classes contain words which are clearly close in meaning. For example, one class contains, among other words :

number amount value figure level rate

Another class is filled with institution-like or group-like nouns :

board body committee party company council government police society industry war trade market office building church school country world family movement

Another contains, among others :

age development education experience knowledge history

There are many more minor semantic similarities in this figure. For example, some antonym pairs are classed closely together : ⟨day⟩ and ⟨night⟩; ⟨morning⟩ and ⟨evening⟩; ⟨boy⟩ and ⟨girl⟩; ⟨man⟩ and ⟨woman⟩; ⟨individual⟩ and ⟨particular⟩ (or even ⟨general⟩ and ⟨individual⟩); ⟨public⟩ and ⟨private⟩; ⟨life⟩ and ⟨death⟩; ⟨water⟩ and ⟨land⟩; ⟨father⟩ and ⟨mother⟩; ⟨men⟩ and ⟨women⟩. Figure 5.20 shows a similar arrangement of words, though not as well defined as those in the previous figure. Several synomyms and antonyms can be identified; reflexive pronouns tend to cluster; numerical amounts cluster; also, plural time measures cluster.

The next set of words to be differentiated have features which are more verb-like than noun-like. They are displayed in figures 5.21 and 5.22. In figure 5.21, there is a major syntactic division between verbs, in one half of the sub-tree, and modifiers in the other half. Within the verb half, there is a broad distinction between base forms of common verbs and

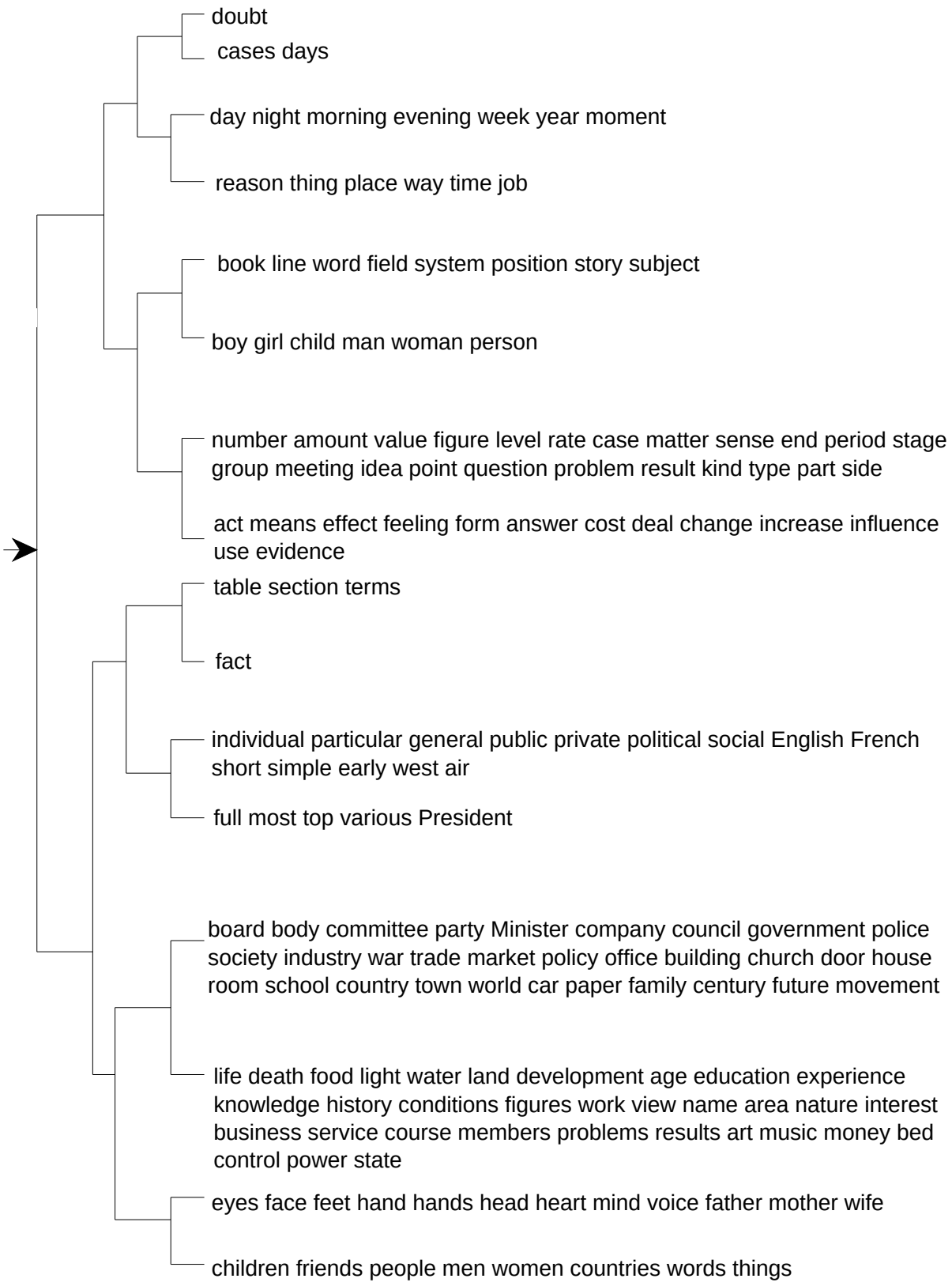


Figure 5.19: Detail, from level 5 to level 9, of many noun-like words. Clear semantic differences are registered.

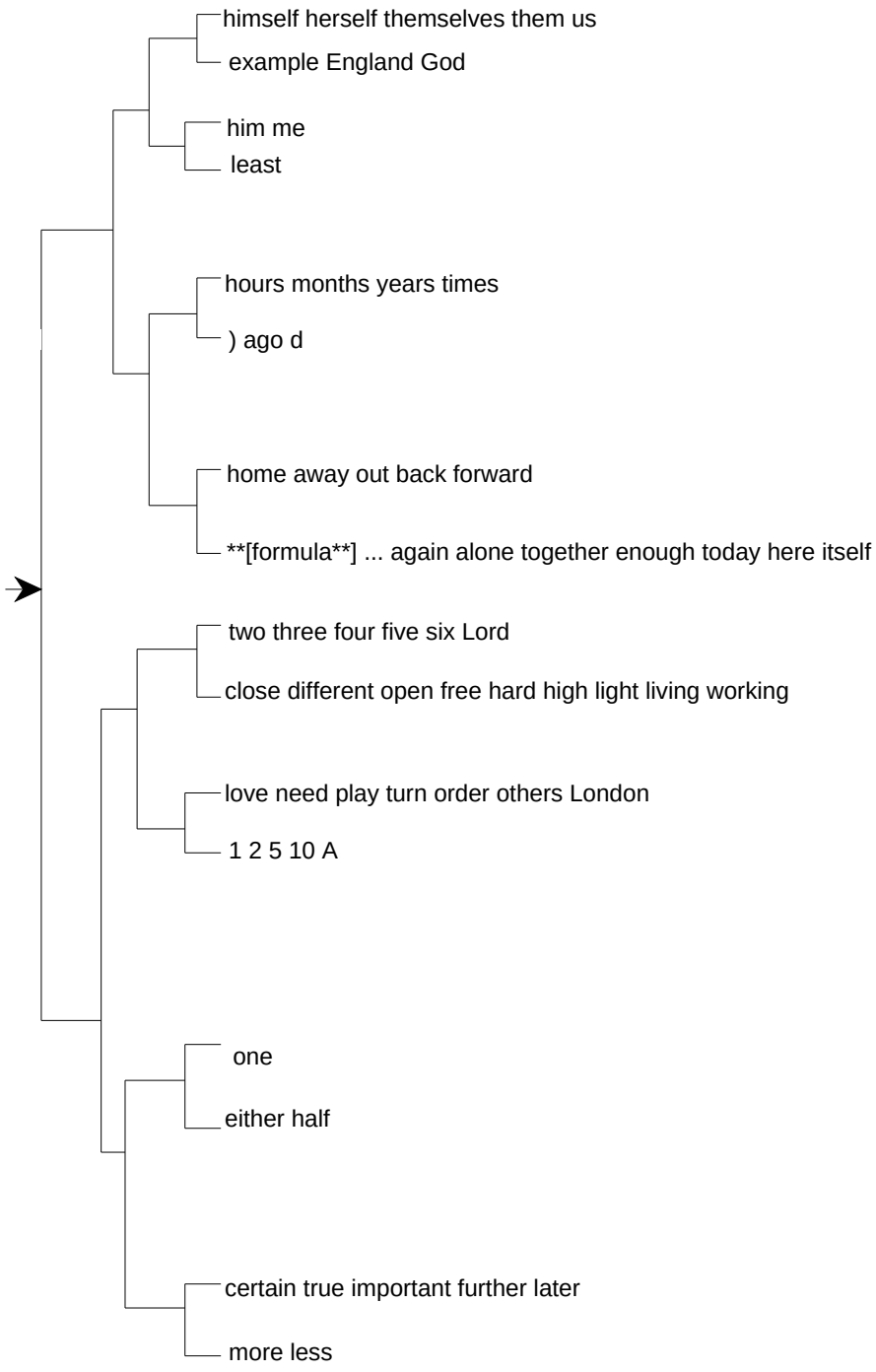


Figure 5.20: Detail from levels to 9 of classification area 00101.

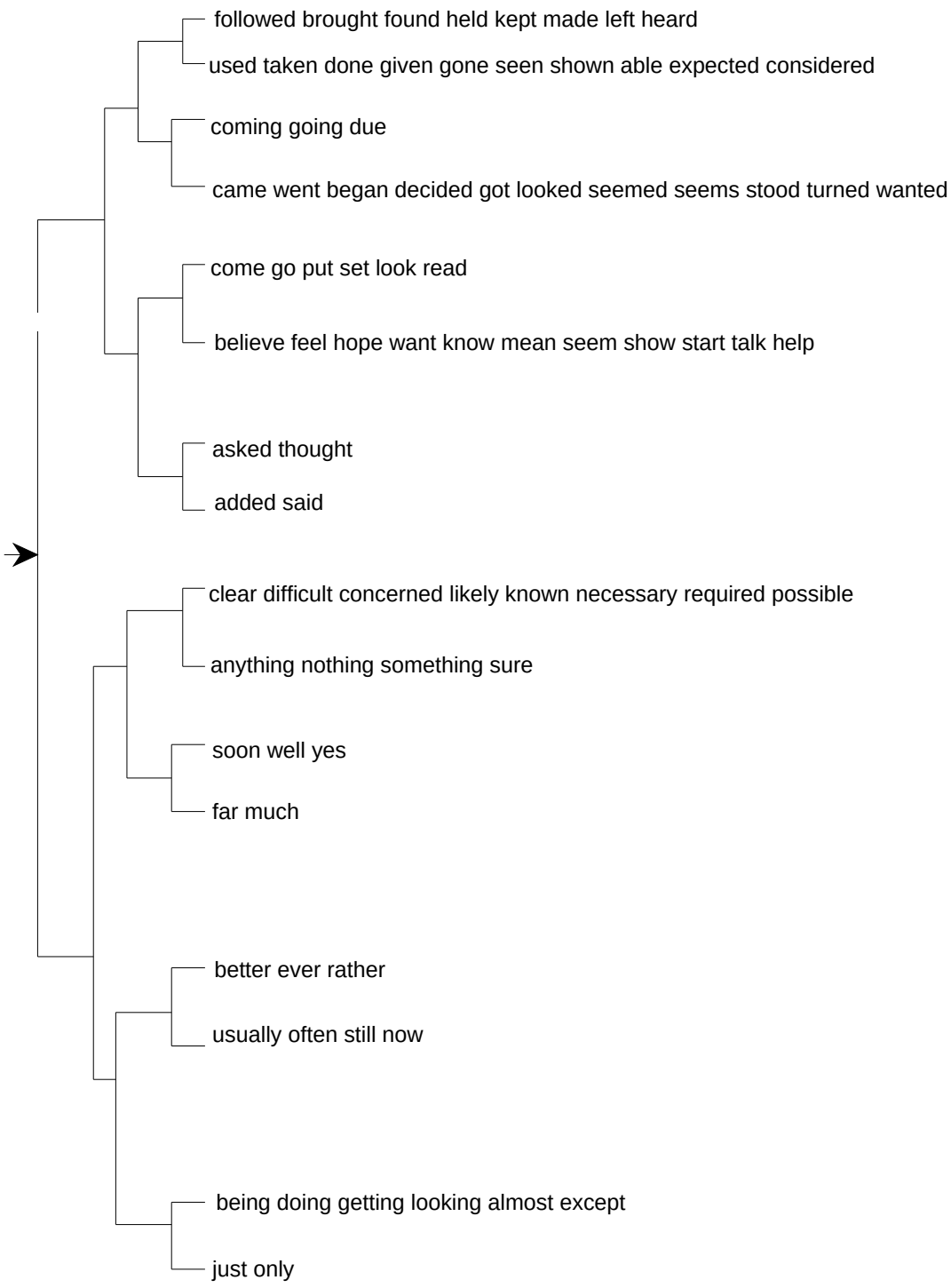


Figure 5.21: Detail from levels to 9 of classification area 00110.

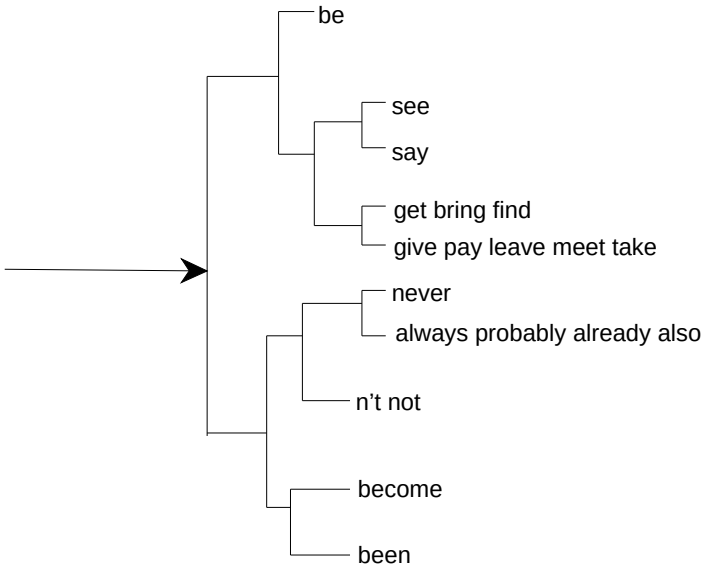


Figure 5.22: Detail from levels to 9 of classification area 00111.

past forms; an exception is the cluster of past tense verbs associated with reporting utterances, which is grouped with base forms. Within the class of past forms of verbs, another distinction is made between past participles :

```
taken done given gone seen shown
```

and verbs whose past participles are the same as their ordinary past tense :

```
followed brought found held kept made left heard
```

The sub-classification of base forms of common verbs has an interesting semantic sub-classification. One of the classes contains verbs concerning mental activity :

```
believe feel hope want know mean seem
```

A nearby class contains the semantically related :

```
come go put set
```

In the other half of the tree shown in figure 5.21, modifiers appear — adjectives and adverbs. As well as the interesting class :

```
anything nothing something sure
```

The adverbs :

```
usually often still now
```

form another semantically related class.

In figure 5.22, the main verb-modifier distinction is observed again. The verbs in this region are all base forms and again their sub-divisions reveal semantic similarities. The modifier-side of the tree contains mostly adverbs.

Figure 5.23 contains many prepositions. It also contains co-ordinating conjunctions (compare with figure 5.18). There are five main preposition clusters in this part of the word taxonomy. The first contains prepositions which describe a peripheral relationship :

```
along about around near over outside
```

The second class contains prepositions which describe a supporting relationship :

```
with behind against upon
```

The third class includes surrounding prepositions : ⟨among⟩ and ⟨between⟩. The next class is similar to the surrounding prepositions; it contains prepositions which mostly describe an internal relation : ⟨in⟩, ⟨during⟩ ⟨within⟩ and, less convincingly, ⟨under⟩. The final class of prepositions describe an over-arching relationship : ⟨on⟩, ⟨through⟩, ⟨across⟩, ⟨into⟩ and ⟨towards⟩.

Interestingly, the word ⟨de⟩ gets grouped with ⟨of⟩. A quick analysis of the LOB corpus reveals that most of the occurrences of ⟨de⟩ are as the French word meaning ⟨of⟩ — in, for example, segments like

```
Charles de Gaulle
```

The final figure in this analysis is figure 5.24. There are three interesting sets of words in this part of the vocabulary classification. Firstly, modal verbs cluster closely; secondly, close to modals, various parts of the common verbs ⟨be⟩, ⟨have⟩ and ⟨do⟩ cluster. Together these two groups capture most of the common verb auxiliaries. The third main group contains mostly adjectives. Finer sub-divisions also reveal semantic commonalities — for example one region of the tree contains words like :

```
big great large real special personal small little
```

Another contains :

```
best first present next last
```

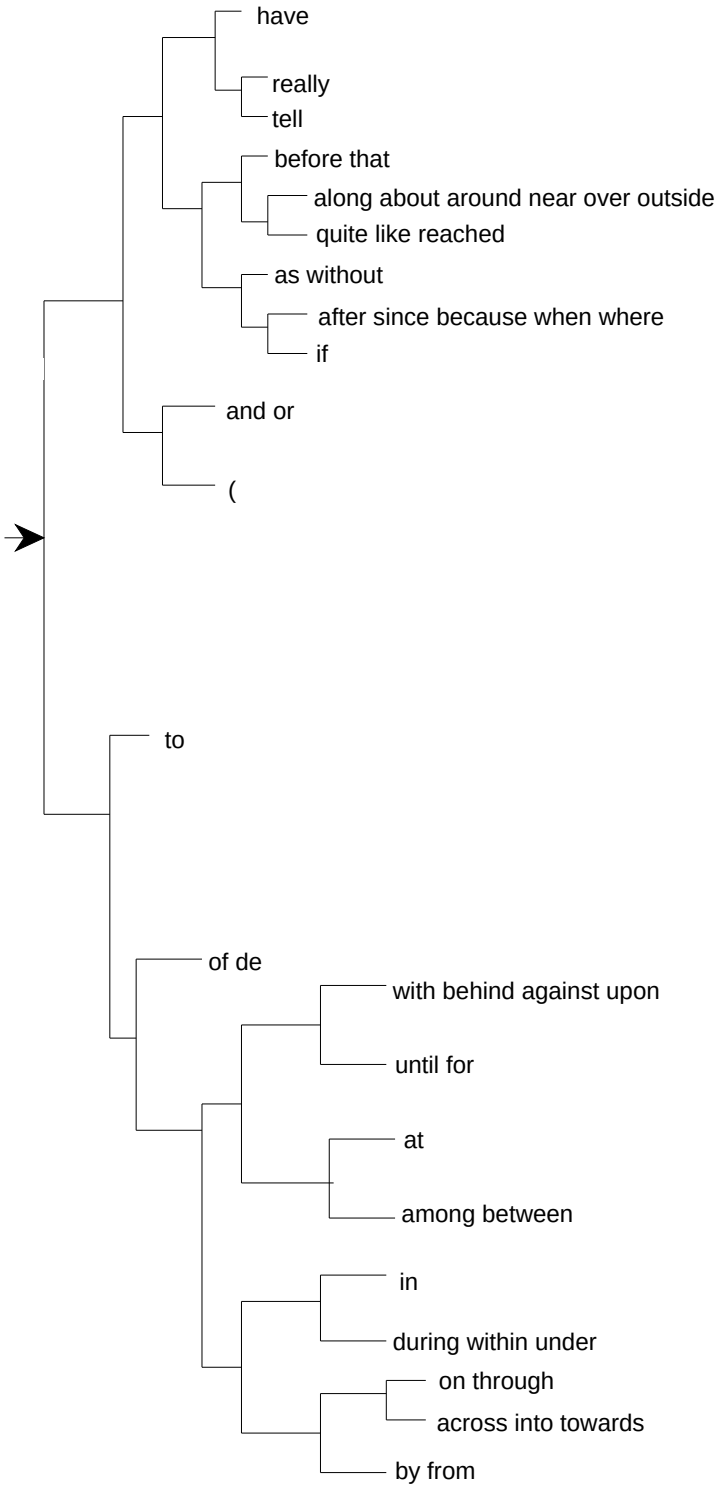


Figure 5.23: Tree containing most common prepositions and their inter-relationships.

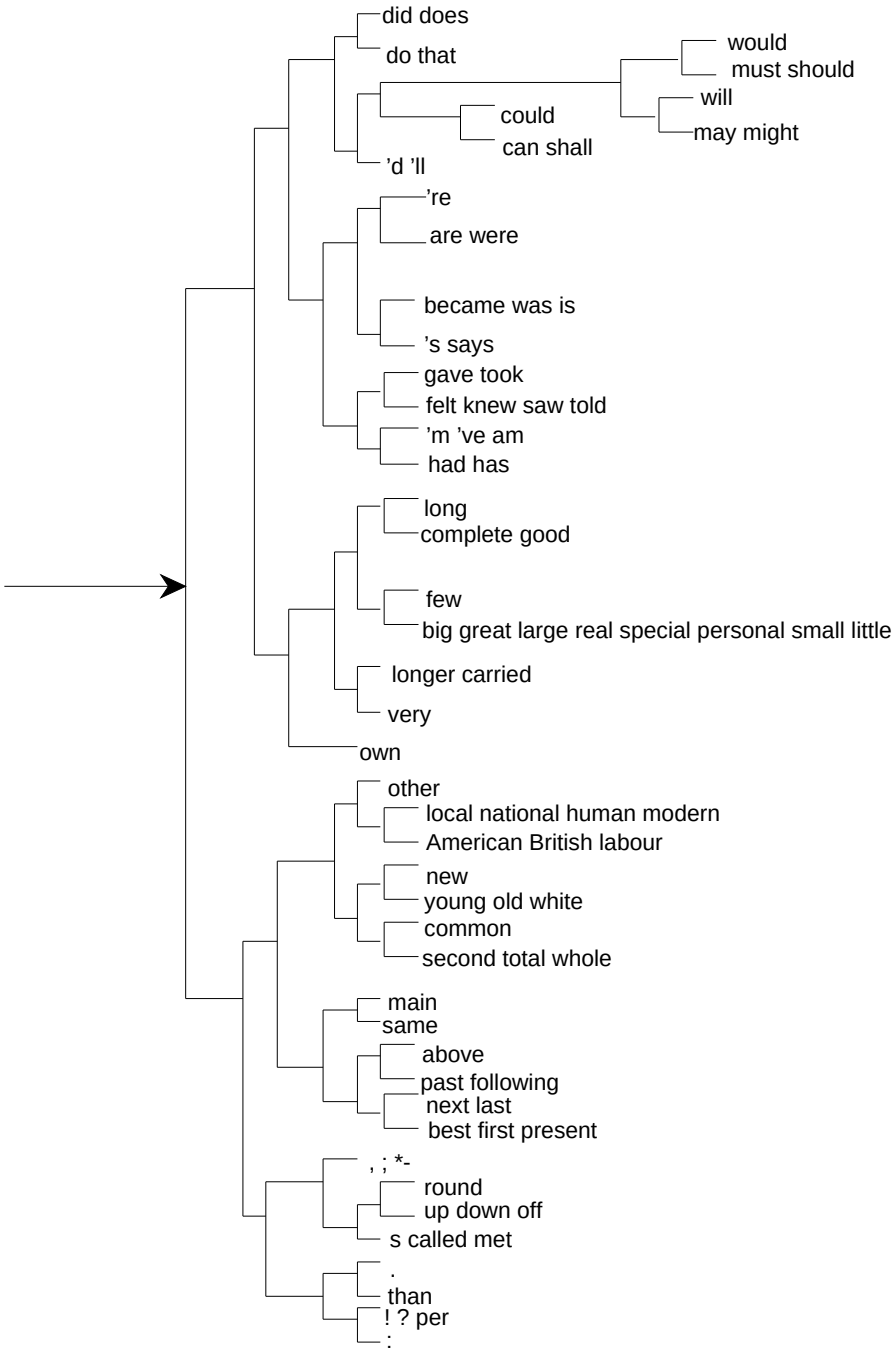


Figure 5.24: Area of word classification showing modal verbs, adjectives and parts of the verbs ⟨be⟩, ⟨do⟩ and ⟨have⟩.

5.6 Clustering of Low Level Linguistic Phenomena — Letters and Phonemes

Cohen [35] states that there are at least two linguistically interesting ways to classify phonemes — one based on acoustics and one based on distributional properties. He quotes G.L. Trager, who asserts :

> Pour arriver à une classification phonématique il faut prendre tous les phonèmes, en totalité, et les examiner de point de vue de leur fonctionnement. De cette façon on arrive souvent à une classification qui est à peu près la même que la classification phonétique, mais elle est *a posteriori* et donc scientifiquement correcte. Toute autre manière de présenter les choses est un procédé à l'envers.

It should be added that a distributional classification of phonemes may not be identical to an acoustic one since distributional classifications should also take account of co-articulation phenomena. Arriving at a digital phonemic classification by examining *only* the acoustic properties of isolated phonemes will not uncover these phenomena. Also, any analysis of transcribed phonetic utterances is bound to include transformations, some of which can distort the acoustic data in significant and undesirable ways.

Figures 5.25 and 5.26 show the results of clustering on a phonemically transcribed version of the VODIS corpus and a letter clustering of the VODIS corpus respectively. The most obvious feature of these results is a successful distinction between vowel related and consonant related phonemes and letters. The distinction is clearer with phonemes, partly due to the non-linear relationship between phonetic and orthographic transcriptions. Section 5.4 mentions some of these distinctions in more detail. Beyond the vowel-consonant distinction, other similarities emerge : vowel sounds with particular vocal tract positions are clustered closely — the ⟨a⟩ sounds, for example, and the phoneme pair ⟨o⟩ and ⟨oo⟩; some consonants which are similarly articulated also map onto local regions of the classification space — ⟨r⟩ and ⟨rx⟩, ⟨ch⟩ and ⟨z⟩ and ⟨n⟩ and ⟨ng⟩, for example.

5.7 Clustering of High Level Linguistic Phenomena

Much information about syntactic word classes can be derived from the primitive bigram statistic. The following experiment investigates how much semantic information resides in such a narrow representation of context. A word-category hybrid version of the LOB corpus was created — any word token which is neither a verb nor a noun is replaced by the syntactic

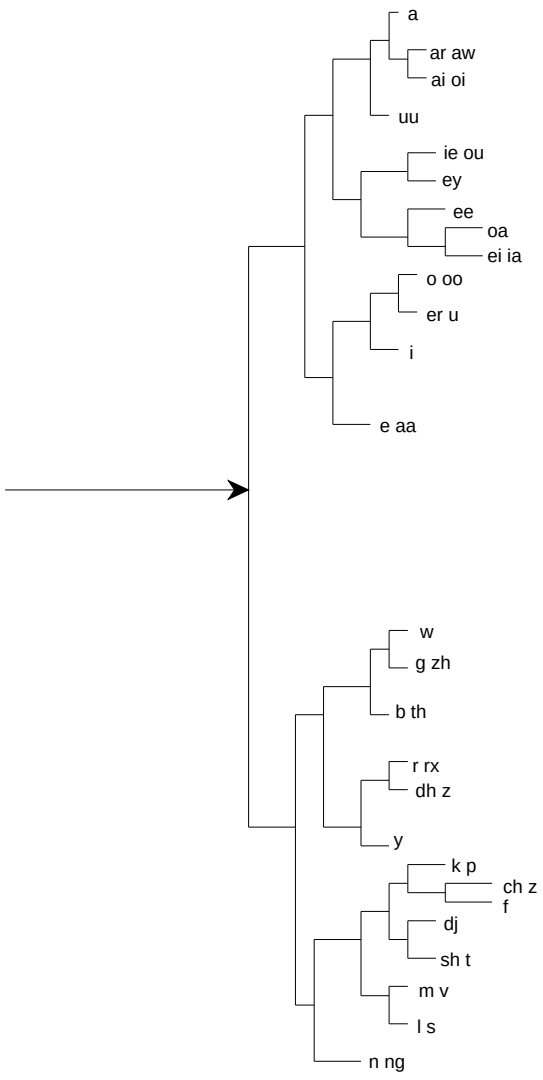


Figure 5.25: Automatic Phoneme Clustering which differentiates between vowels and consonants

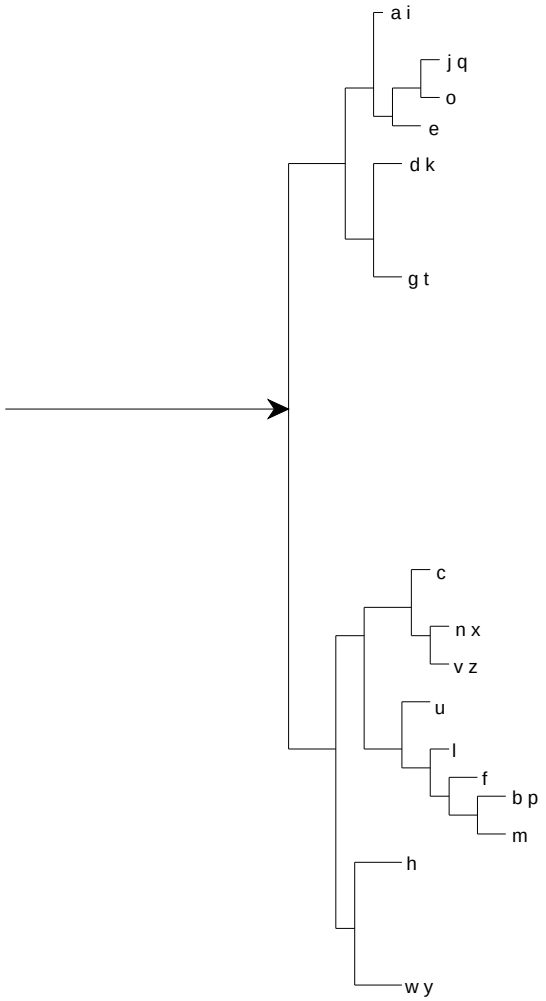


Figure 5.26: Automatic Cluster Structure of Letters from VODIS corpus — the vowel-consonant distinction is less marked.

class which is associated with it in the original corpus. Only singular nouns and third person singular and base forms of verbs were considered — that is, words remain as words only if their associated LOB category was 'NN', 'VB' or 'VBZ'. A total of 3,025 hybrid cluster elements were selected for the experiment, under 200 of which were word classes. Unfortunately, time limitations prevented a more detailed study than the one now given, although some clear semantic links suggest themselves. Some of the more interesting results are shown in figures 5.27 and 5.28, which list some word classes extracted from level 9. Of the 512 classes at this level, only 38 of them have a membership of greater than ten words. Thirteen of these classes are shown in the two figures, as well as 4 other semantic classes of size less than 10. Each box in the figure contains all of the words from that class — no anomalous words have been excluded. However, the word order within each class has been manipulated to highlight their semantic qualities. In some groups, especially the larger ones, we suggest that at lower levels of classification more semantic distinction will be uncovered. The remaining 25 classes not reported here also exhibit a great deal of semantic clustering. Of course, all of these classes exhibit some spurious word classifications.

5.8 Generalisation to Other Languages

In order to test if the clustering algorithm discovers structure in other languages, an experiment was carried out with the 281 most frequently occurring words from a computer readable version of the complete works of Cicero, supplied by the Oxford Text Archive. The results of this are shown in figure 5.29. Latin words have more morphological complexity than English words, yet the figure does reveal some syntactic classes; for example, the class of words circled in the figure are mostly prepositions. Many other clusters of closed class words are present, although the overall structure is quantitatively much more fragmented than a comparable English-based classification. This is to be expected : the nominative case of a noun should have a measurably different distribution to the genitive of the same noun.

As with the English language, prepositions get clustered closely together :

```
me nobis nos se te vobis vos
```

Negators cluster :

```
nemo nihil non nulla nullum
```

as do the related adverbs :

```
post quantum quid quis quod ubi
```

Figure 5.27: Semantic clustering results. Memberships of 11 of the 38 classes whose size is greater than ten, at a classification level of 9 bits. From the LOB corpus. Body parts, relatives, mental states, human roles, house parts, two classes of mental verb, relation verbs, hope-nouns, effort verbs.

admiration affection agony anger ambition amusement anticipation anxiety apathy authority awareness beauty belief bitterness chaos character coincidence communication composition continuity concentration concern confidence conflict conformity controversy confusion consciousness courage courtesy criticism cruelty depression despair destiny devotion difficulty dignity disappointment disorder disturbance drama efficiency enjoyment entertainment enthusiasm equality evil excitement experience failure fairness fantasy freedom friendship frustration fun genius goodwill gossip grace gratitude gravity grief growth guilt happiness harmony haste hatred hospitality humanity humour ideology ignorance impatience impulse inclination independence indignation injustice inspiration interest jealousy joy judgement judgment justice kindness laughter learning liberty licence luck maturity morality nonsense optimism pain patience perception personality perspective philosophy pleasure popularity pregnancy prejudice prestige pride principle quality questioning rage realism relief resentment respect responsibility rivalry romance salvation satisfaction scholarship schooling shame shock silence sin size skill stability style success symmetry sympathy talent temperament tension terror tiredness tragedy triumph trouble uncertainty urgency victory vigour violence virtue warmth weakness wealth weight spirit communism democracy socialism federation parliament technology tradition revolution evolution progress nature society literature poetry knowledge thinking singing writing thought language logic mathematics disarmament politics economics economy investment inflation employment productivity policy pornography poverty publicity reality religion hell paradise theology action effect activity routine output production occupation motion movement operation occurrence behaviour capacity essence shape being baptism life death suicide punishment execution disease illness injury medicine destruction disaster imprisonment residence settlement adultery sex marriage crime warning error myth after-care agriculture admission access exposure convenience addition accumulation expansion consumption possession excess completion duration consequence contamination contact greeting assistance evidence information example selection explanation observation interpretation instruction behalf detail attention consideration conversation consultation permission investigation inspection supervision view visibility sight discussion advantage celebration ceremony tribute accordance preparation recognition protection redemption remuneration provision pursuit adjustment approximation limitation precision compromise negotiation analogy correspondence comparison parallel practice contrast response reference relation rhythm conjunction connection connexion integration implication suspicion rumour outlook direction expulsion acid antibody radioactivity oxidation chemistry physics dose part thread bottle sweat fluid liquid gin sherry whisky wine bream cap cream toast gear material money payment hay vegetation wheat wood wool rose acre pitch place hole layer length age date midnight instance annum white black mist chairman president decoration painting firing noise sport

appendix diagram fig p Op page paragraph momentum heaven suet

aluminium cotton gold hydrogen oxygen polythene sodium grill salvage football tennis prison wartime grading

boiling baking cooking engineering manufacturing shopping housing boarding trading export finance finishing sintering electricity ignition motor rocket railway science-fiction bismuth brass metal nerve oak resin grammar noun infant kinship dairy census county amateur golf

bending torsion-flexure drift equilibrium equity rotor coolant ion exhaust humus leather plastic textile consumer retail labour wage borstal constituency qualifying slenderness

bend boil brush clasp cover cut feed fly grasp help hurt kick kiss knock melt move offset open pass pop push register repair repeat ride ruin run sail scream sleep stand start stay stick swim talk tie turn wake walk wash watch wear worry blame regret call concentrate look count fancy offer guess mention

Figure 5.28: More semantic clustering results. Memberships of 6 of the 38 classes whose size is greater than ten, at a classification level of 9 bits. From the LOB corpus. Mental states, writing reference, materials, materials and processing, more materials and processing, common manipulation verbs.

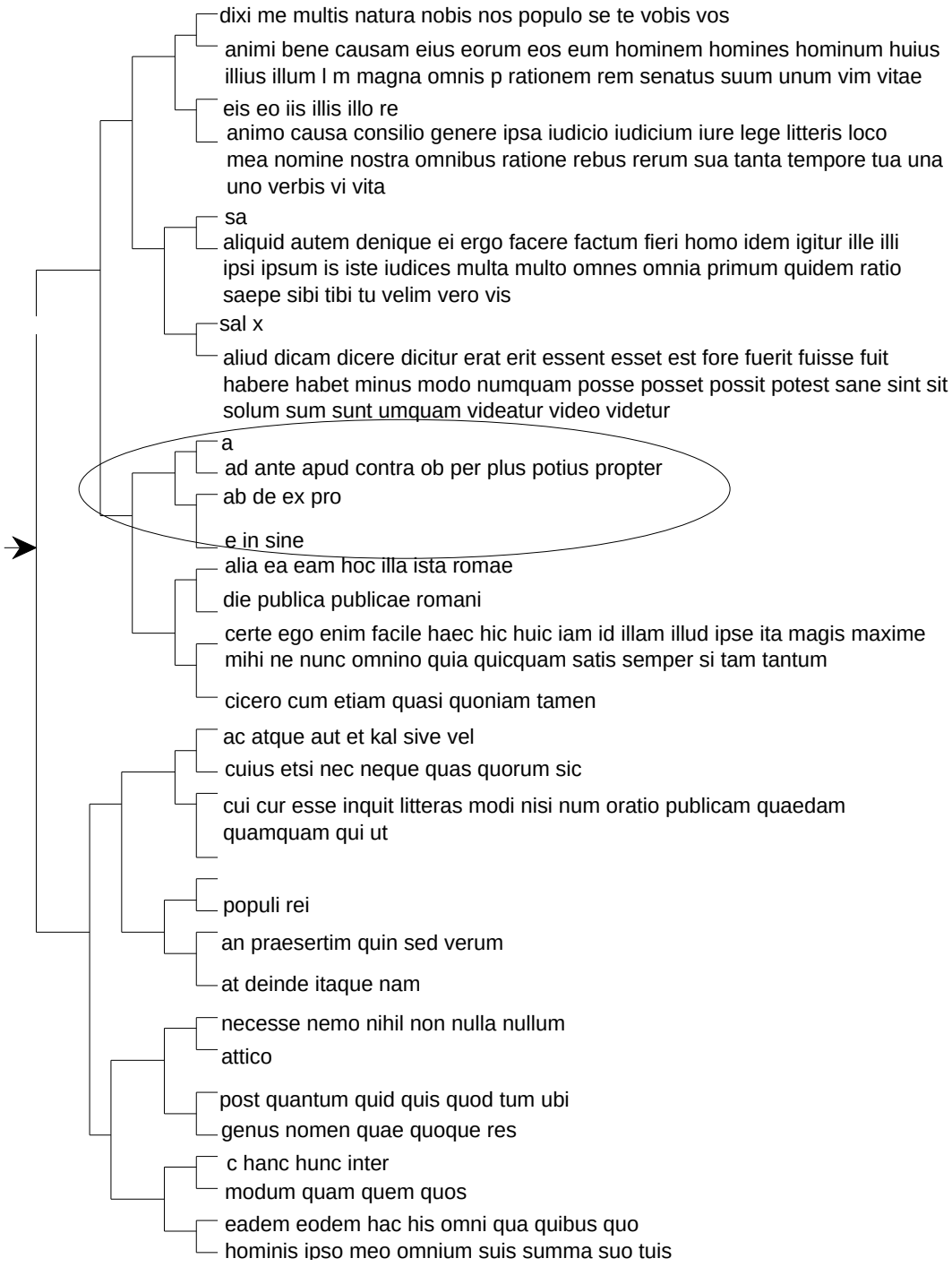


Figure 5.29: Classification of the most frequent words in a formatted version of the complete works of Cicero, in Latin; a group of prepositions is highlighted to shown that the clustering system can find structure in languages other than English.

and reflexive pronouns :

```
ipso meo suis suo tuis
```

5.9 Experimental Conclusions

The experiments described above demonstrate the capacity for a simple version of the average class mutual information maximiser to discover some of the structure of language. They also highlight the dependence of automatic word classification systems on corpora with sufficient contextual variety. Elman hoped to show that linear linguistic information could implicitly contain hierarchical structure. His further hope that this demonstrates how it is possible to describe language without recourse to the traditional prior structures of the linguist has not been fully realised. Chomsky's claim about surface features not providing sufficient information to induce grammar was applied to natural languages, which are arguably context free. Elman's grammar is finite. Similar experiments need to be performed on a language which is context free in order to approach a challenge to Chomsky's hypothesis. The automatic classification using structural tags described in this chapter has some advantages over other systems. Unlike many neural net classifiers, this system has an explicit hierarchy, ready for use with statistical language models. Also, it involves ideas which have strong theoretical support in the field of information theory — section 4.4 expand some of these connections. Among information-based word classification systems, the current model has at least three advantages over, say, the system described in Brown *et al.* [18] : it allows immediate, simultaneous access to all levels of word classification which are available to a language model; secondly, it makes word-classification decisions at each level which can be (and indeed *are*) later reversible; thirdly, it offers a more appealing way of representing the relationship between words and classes, rejecting the traditional view of describing the relationship between a word and its class as a function between two sets of objects in favour of a more structuralist conception, where the word's representation itself contains all of the information needed to establish its class.

The experiments lead to the conclusion that large corpora are better for word classification because the set of syntactic contexts of a given word swamp its semantic contexts; in this ideal case, broad level divisions are made on syntactic grounds, while fine-grained divisions occur for semantic reasons. Jelinek *et al.* [86] and Sampson [140] provide good grounds for using less-than linguistically perfect classifications in language modelling; their positions are summarised in chapter 3.

The experiments also show that one structure-finding scheme can recover useful information at three major levels of language — phonetic, syntactic and semantic. It has also been demonstrated that the system is not limited to one language; this should be useful in identifying the word classes for languages whose paradigmatic relations have not been well described by linguists; the method also provides a quantitative way of deriving classifications, against which various hypothesised classifications may be compared.

Chapter 6

Evaluating the Clustering Capacity of Mutual Information

6.1 Overview

The previous chapter tried to illustrate the scope and potential of the clustering algorithm; in this chapter, comparisons are made with several recent word classifiers. Important issues are raised, including the problem of making direct comparisons of classifications and discussing the differences in emphasis present in work in this area. Some internal limitations of the system are also presented.

The following section includes a discussion of the difficulties involved with quantitative and qualitative comparison. After that, some attempts are made to measure the relative performance of the system — firstly, the output of the annealing algorithm is compared directly with results published recently in the literature. Next, comparisons are made between the annealing algorithm and two alternative clustering strategies where as many of the experimental details as possible are held constant. The first of these alternatives is the Elman system described in chapter 5; the other is a merge-based classification approach which Brown *et al.* exploit. A version of this system is implemented and presented with the same data as the annealing system. The third type of comparison involves a measure of performance recently introduced by Hughes; this cluster evaluator is implemented and we report results from a benchmark experiment; this allows a comparison to be made with Hughes' own classification system and also that of Finch.

Next, some of the limitations of the annealing system and the structural tag representation are explored. After that, the topic of indirect evaluation is introduced — this provides a context within which chapter 7 can be read.

6.2 Classification Comparison Problems

The best known word classifiers to date are those of Finch and Chater [51, 53], Hughes and Atwell [79, 80], Schütze [143] and Brown *et al.* [18]. Other powerful and impressive systems are described in Brill *et al.* [14], Elman [45] Kneser *et al.* [92] and Pereira *et al.* [123]. The word classification system described here appears to rank among these in performance.

Successful word classification systems are still rare; consequently many researchers spend their research time developing their own systems rather than making comparisons with others systems. A noticeable exception to this pattern is the work of Hughes *et al.*, who have developed a way to compare classifications if the original corpus has been pre-tagged in a linguistically sensible way. They criticise the most common approach to classification comparison — that of qualitative comparison — as too reliant on the preconceptions of the researcher; instead they suggest that the intuitions of linguists (computation or theoretical), encoded in the syntactic categories which some tagged corpora possess, afford a more methodological approach. This is a significant research contribution. Brill and Marcus [15] have also developed a simpler version of this benchmark approach to classification evaluation; they use the Penn treebank version of the Brown corpus and, as with Hughes, a reduced tag set.

A scientifically reliable method of comparing classifications would be to measure how different they are from randomly generated classifications. This kind of approach has been taken by Redington [132] but is not used here because it is assumed that the classifications are clearly different from random ones, and secondly because many classification processes could produce distributions which are non-random, but which have nothing to do with lexical categories — for example, a classification could be based on alphabetical order. However, random classification experiments could be useful as baseline values in any quantitative system.

The question of the criterion of a successful classification is dependent upon research motivations, which fall into two broad schools : those who primarily would like to recover the structures which linguists posit — these structures are mainly syntactic but also partly semantic. The second school is interested in classifications which help to improve some language model or other language processing system and which may or may not exhibit linguistically perspicuous categories. Unless modern linguistics is radically wrong, a large degree of overlap should occur in these two sets of ideal classification.

If a researcher claims that linguistically well-formed classifications are not the immediate goal of their research, they must find some other way of measuring the applicability of their classifications. This indirect evaluation is introduced in the next chapter — it operationally defines good word classifications as those which confer performance improvements to

statistical language models.

Another difficulty in identifying ideal classifications concerns the fact that many words have multimodal distributional properties; this is the problem of word ambiguity, in traditional computational linguistics. Also, direct comparisons are made more difficult because different researchers use different corpora, and corpora of different sizes. Generally, the more domain-specific a corpus, the better the likelihood of deriving semantic significance, but the greater the danger of encoding the syntactic idiosyncrasies of the corpus; the larger the corpus, the more representative a sample there is of a word's distribution. This is especially true when comparing the results of Brown *et al.*, which are based on a corpus which is comfortably two orders of magnitude larger than the largest corpus used in this research.

The comparison technique described by Hughes *et al.*, based on prior tagging, is useful; however, the evaluation produced by the system is only as good as the tagged corpora upon which the test is based. This is relevant in cases where a classification for a rare foreign language is derived or when a semantic classification is derived, and where no properly tagged benchmark corpus exists. Fortunately, this is less of a problem with purely syntactic classifications — the LOB corpus is syntactically tagged and is large enough to provide a benchmark evaluation for classifications.

6.3 How the System Compares with Others

Each of the following subsections highlights some comparison between the current system — referred to as SAMI, simulated annealing and mutual information, for brevity — and others described in the literature. The aim is to provide as encompassing an evaluation of different automatic word classification systems as possible.

6.3.1 Direct Qualitative Comparison

Qualitative comparison is the least attractive way of evaluating the success of a classification algorithm; unfortunately it is used in some cases since no practicable alternative exists.

In Brill *et al.* [14], another automatic word classification algorithm was developed and trained using the Brown corpus; this is approximately the same size as the LOB corpus, so any difference in results between the two algorithms is less likely to be due to corpus effects and more likely to be related to the quality of the algorithm.

Firstly, Brill *et al.* rely on contiguous bigrams only : that is, bigrams of the type $\langle x_i, x_{i+1} \rangle$; they report success at partitioning words into word classes. They note that pronouns have a

disjoint classification, since the `+nominative` and `-nominative` pronouns — *e.g.* ⟨`I`⟩, ⟨`they`⟩ and ⟨`me`⟩, ⟨`them`⟩ respectively — have dissimilar distributions. These effects are replicated in our experiment, described in section 5.5 : `+nominative` pronouns are found in a region of classification space shown in figure 5.16, and most of the `-nominative` pronouns are found in the area shown in figure 5.20. Exceptions to the clustering of `-nominative` pronouns are ⟨`you`⟩ and ⟨`her`⟩; ⟨`you`⟩ is the form of the `+nominative` *and* the `-nominative`, and 'her' is also a possessive determiner. In both these cases, the alternative form determines the word's final classification.

They report other, more fine-grained features such as possessives, singular determiners, definite-determiners and WH-adjuncts. The SAMI algorithm also distinguishes these features. Brill *et al.* do not report any substantial adjective clustering, or noun clustering, or singular-plural differences, or co-ordinating and subordinating conjunction distinction, or verb tense differentiation. At deeper levels, the only semantic clustering they report involves the group :

```
man world time life work people years
```

and the group :

```
give make take find
```

The SAMI algorithm generates substantially more semantic detail : mental verbs, body parts, social institutions, time measures, human types and amount-nouns. It also generates many synonyms and antonyms. Even prepositions are sub-classified semantically. This qualitative comparison with the work of Brill *et al.* suggests that the structural-tag based algorithm is substantially better than one based on traditional distributional analysis, as designed by Brill *et al.*

The results described in Brown *et al.* [18] are based on a training set two orders of magnitude greater than the one used in the above experiment. Even the vocabulary size is an order of magnitude bigger. As the vocabulary size is increased, the new vocabulary items tend, with a probability approaching unity, to be content words : after approximately one thousand words, few function words are left undiscovered. This increase in resources makes contexts more balanced and, simultaneously, more statistically significant. It also allows many more content words to be grouped together semantically. The authors give two tables of generated word classes, one being specially selected by them and the other containing randomly selected classes.

The list of best classes is impressive; weekdays, months, directions, semantically related modifiers and Christian names each classed separately, and with low levels of mis-classification. They report a class :

```
head body hands eyes voice arm seat eye hair mouth
```

which is similar to one generated by the SAMI algorithm and which is reported in figure 5.19. Their set of randomly generated classes contains a higher level of mis-classification but is still impressive. They do not report on any taxonomic relations between these classes, so it is no possible to compare the broad detail of the two sets of data. We shall return to this merge-based approach later.

The results of Finch and Chater [51, 50] are also based on a substantially larger corpus — 45,000,000 words. They describe the overall structure of their taxonomy and give specific examples of parts of the tree. With a vocabulary of 1,000 words, they can report more semantic relations between content words. The overall structure of their tree is similar to the tree generated in the above experiment; the main elements of the present classification tree are described in figure 6.1 — the Finch and Chater tree is comparable to this. In their tree, the distributions of nouns and verbs do not appear to be as tightly clustered as they are in figure 6.1. Finch *et al.* also run a version of the Elman experiment. Using the same sentence templates, their system fails to produce a complete noun-verb distinction at the highest level, though they offer an argument to suggest that the inadequacy lies in the nature of Elman's pseudo-natural language corpus; the current annealing system uses Elman's corpus but succeeds in making the primary noun-verb distinction. Finch augments the Elman grammar with extra rules, and this modified language provides a corpus which does allow Finch's system to learn the noun-verb distinction. Finch also clusters letters and phonemes. He has complete success in distinguishing between vowels and consonants in the letter experiment, and only the phoneme `/u/` is incorrectly classified in the phoneme experiment. Conversely, the annealing algorithm completely clusters phonemes into vowels and consonants, but performs less well with letters; interestingly, the wrongly classified letter vowel was the letter ⟨u⟩. Finch also reports limited success with deeper semantic clustering. Another difference in the experimental detail of Finch's work compared to the present work, is that sentence boundary information is encoded in the Finch system, whereas in the current system, punctuation markers are considered as words in a constant stream. No firm conclusions can be drawn about the relative merits and disadvantages of these systems — again, later work will allow us to make a direct quantitative comparison.

The results of Hughes *et al.* are also as impressive as the Finch and Chater system —

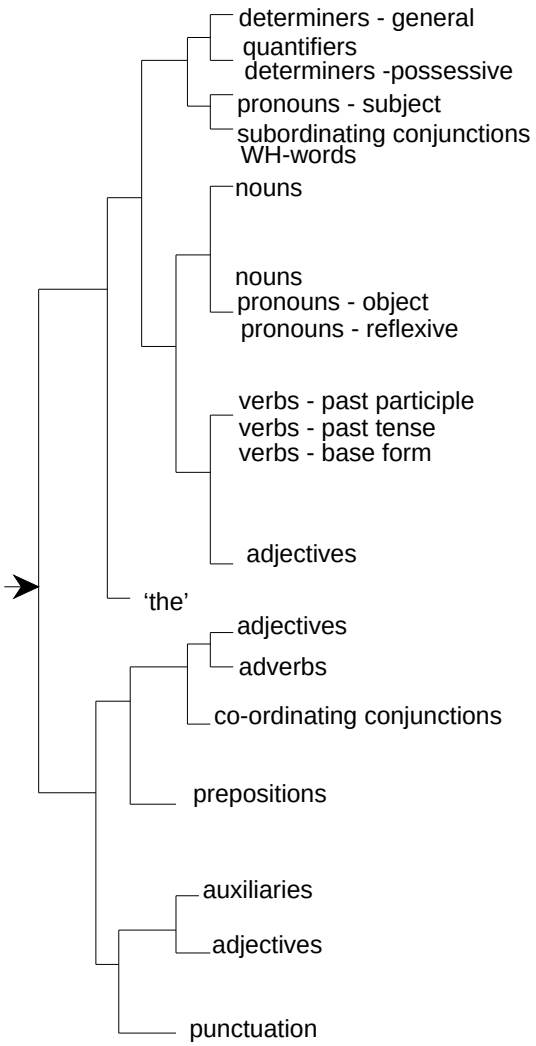


Figure 6.1: Approximate topology of the tree generated from the LOB corpus and the average class mutual information maximiser, using structural tags.

a strong overall structure combined with many definite syntactic and some semantic detail revealed at lower levels of classification.

Pereira and Tishby [123] do not give details of syntactic similarity — they concentrate on a small number of words and make fine-grained semantic differentiations between them.

Schütze [143] uses a standard sparse matrix algorithm with neural networks; his system is the only one which attempts to side-step the problem of deciding what his clusters are clusters *of*, by producing a system which generates its own class labels. Although he does not report the overall structure of his one-level classification, the quality of the classes his system generates are just as impressive as those of Finch and Chater. His training set is on order of magnitude bigger than the largest one used in the present experiments.

It should be re-emphasised at this point that the work of Schütze, Hughes *et.al* and Finch and Chater uses $\langle w_{x-2}, w_x \rangle$, $\langle w_{x-1}, w_x \rangle$, $\langle w_x, w_{x+1} \rangle$ and $\langle w_x, w_{x+2} \rangle$ bigrams whereas the SAMI algorithm currently only uses contiguous bigrams. This is the most important difference between the current system and these three systems.

6.3.2 Ceteris Paribus Qualitative Comparison

This section describes comparisons which are made between the SAMI algorithm and other algorithms, where some of the parameters associated with the different systems are controlled — for example corpus size.

A Recurrent Neural Network and a Finite State Grammar

The first balanced comparison between the current system and another reported in the literature has already been described in section 5.3. An examination of the results from the Elman experiment and the simulation using the same grammar and corpus size — shown in figures 5.8 and 5.9 — shows that Elman's recurrent network is stronger locally, but weaker globally — the two systems have complementary strengths.

Classification using a Merging Algorithm

The systems described in Brown *et al.* and Brill *et al.* [15] both provide examples of bottom-up, merge-based classification systems; a version of such a system was chosen to be implemented and tested against the SAMI algorithm, using the same input data. The Brown system uses a principle of *class merging* as its main clustering technique. The initial classification contains as many classes as there are words to classify, each word in its own class. Initially these classes are all mutually independent. Then two classes are chosen to merge; the crite-

rion of choice is based on a mutual information calculation. The process is repeated until only one class remains. Next, the *order* of merging provides enough information for a hierarchical cluster to be constructed.

We implemented a version of the system using a pair of data structures, one containing fixed word unigram and bigram frequencies, and another, which is updated after each merge is made, containing unigram and bigram class frequencies. Initially, these two structures are identical — corresponding to the case where each word is the only member of a unique class. Another data structure was maintained to contain a list of remaining classes. After a merge decision has been made, the word members of one class are moved into another class; the first class is then removed from the list of remaining classes. These data structures are not enough, however, to generate a full hierarchical classification after the process finishes, since no record of sub-classifications is kept as the merge process continues. A fourth data-structure contains at any time a bracketed list of the words which currently belong to that class; if this class merges with a second class, the original word list for one of the classes is replaced by a bracketed pair consisting of both member sets (the second class is chosen, arbitrarily). Thus, for example, if class 1 contains the word `five` and class 2 contains `six`, then after the merge, class 1 will be empty and class 2 will contain the string `(five six)`. If class 2 now merges with class 57, which contains `((seven)eight)`, then class 2 will be empty and class 57 will contain the string `((five six)((seven)eight))`. When only one class remains, the data structure containing these strings will have one element from which the entire classification can be generated.

A comparison experiment was designed using the VODIS corpus as a source of frequency information; the annealing system and the merging system were given a set of those words from the de-capitalised and de-punctuated corpus[1] whose frequencies were greater than 30. This accounted for the 256 most frequent words.

The final classifications, to a depth of five levels, are shown in figure 6.2 and figure 6.3 for the merge-based and annealing-based systems respectively. The difficulty of comparative evaluation becomes more apparent when looking at these two classification techniques; both seem to have strengths and weaknesses — for example, the annealing system possesses a more balanced overall structure, compared to the merge-based system, which has as its most important distinction the difference between number-words and all other words. It should be stated that both systems successfully discover a large degree of syntactic structure and some semantic structure also, given the relatively small size of the corpus : prepositions,

[1] except for the apostrophe when it is a part of a word

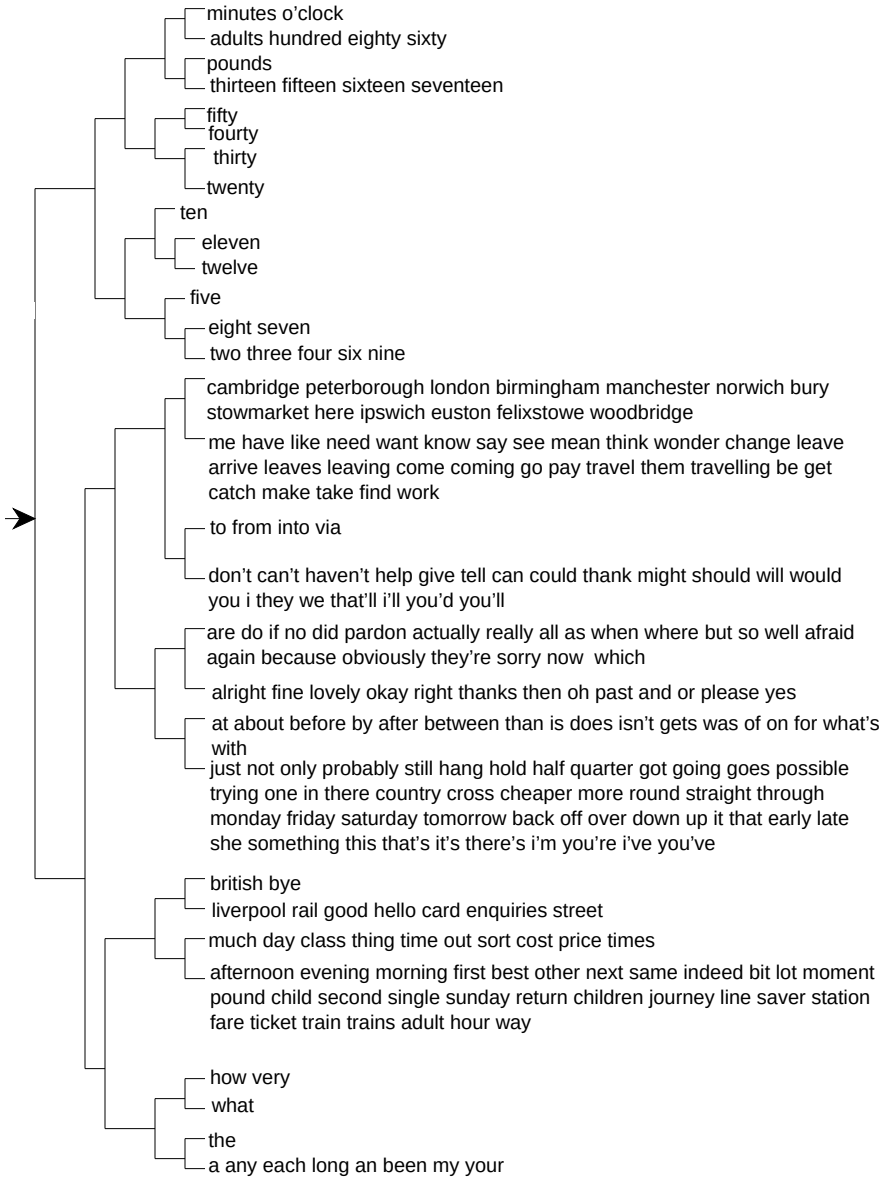


Figure 6.2: Classification of the most frequent words of a formatted VODIS corpus, using a merge-based method.

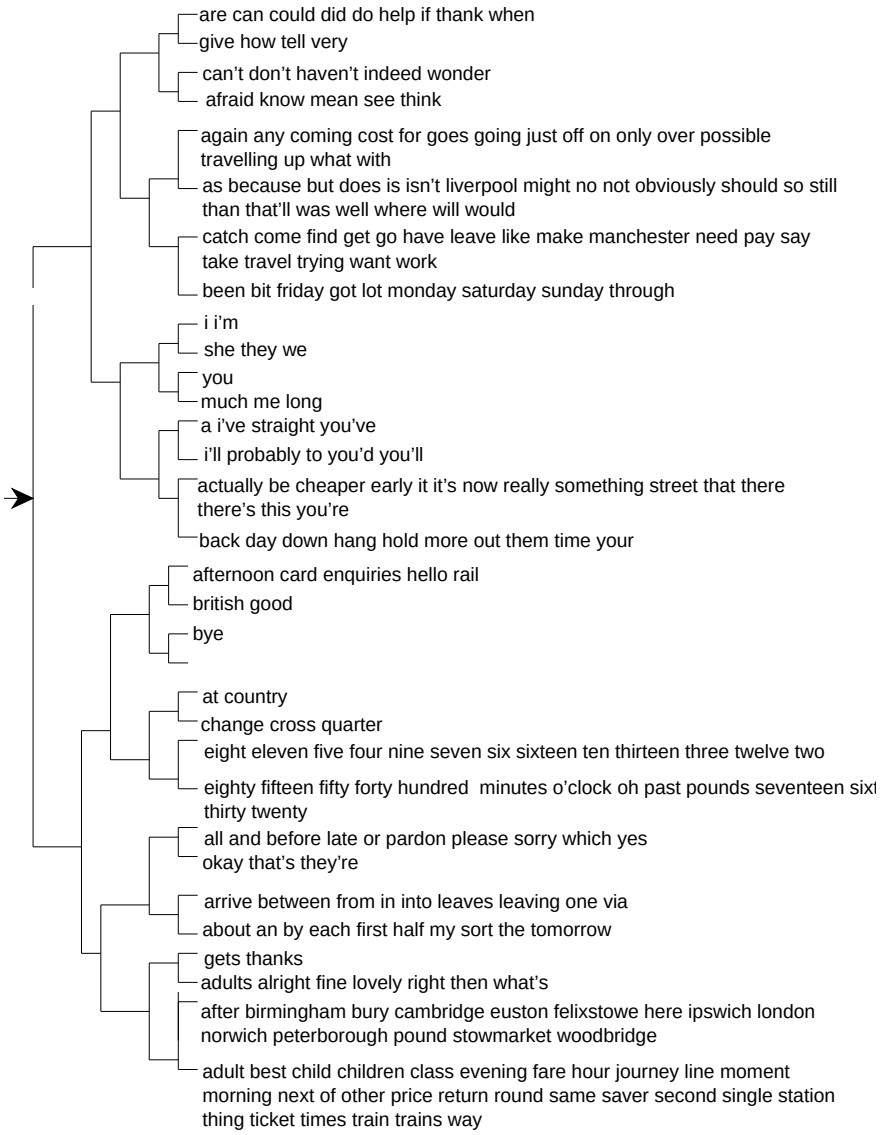


Figure 6.3: Classification of the most frequent words of a formatted VODIS corpus, using an annealing-based method.

determiners, nouns, verbs, adverbs, and conjunctions, as well as place names and number-words.

In order to examine the relative strengths and weaknesses of the two systems, a clearer picture of the distributions of major syntactic categories needs to be provided. These are reproduced in figures 6.4 to 6.9 — in each figure, only those words which unambiguously belong to a specified syntactic category are shown, to a depth of five levels, as before; in this form, the degree of spread can more easily be judged.

The first pair of related-word classifications is shown in figures 6.4 and 6.5; here, the distribution of the most frequent nouns is shown. The globally loose quality of the merge-based system compares slightly unfavourably to the annealing system : the former has 39 and 44 nouns in each half of the classification tree, whereas the latter has 11 and 72 in each half; that is, the annealing method has a tighter overall noun cluster. The figures also show that both systems identified common place names as being lexically similar; the degree of similarity between groups of words is reflected in how tightly they bind to each other at finer grained levels of classification. This allows us to differentiate between the contrasting ways that the number words are clustered. While both systems identify this group as being well-defined and homogeneous, the annealing system clusters them together even at level four, whereas the merge-based method binds them together only as far as level 1; by level 5, it has fractured the classification into 14 sub-classes. In the annealing system, place names are close to the other large noun-class, whereas in the merge-based system, they are at opposite ends of the level 2 classification sub-space.

In figures 6.6 and 6.7, details of the verb distribution are highlighted for each clustering method. In this case, the merge-based system produces a narrower dispersion of words. However, this success is slightly moderated by the consideration that one half of the entire merge classification system was allocated to number-words, leaving the rest of the vocabulary, including the verbs, to be distributed through the other half; even so, the merge-based distribution of verbs still forms a more coherent cluster. In figures 6.8 and 6.9, details of the distributions of pronouns, prepositions and determiners are given. The annealing method produces a better pronoun distribution, but in the case of prepositions and determiners, the merge-based system delivers a more syntactically relevant classification.

In conclusion, the two systems possess a similar set of feature differences as the Elman and SAMI algorithm — that is, the annealing method slightly outperforms with respect to overall classification structure, but looses in quality at lower levels.

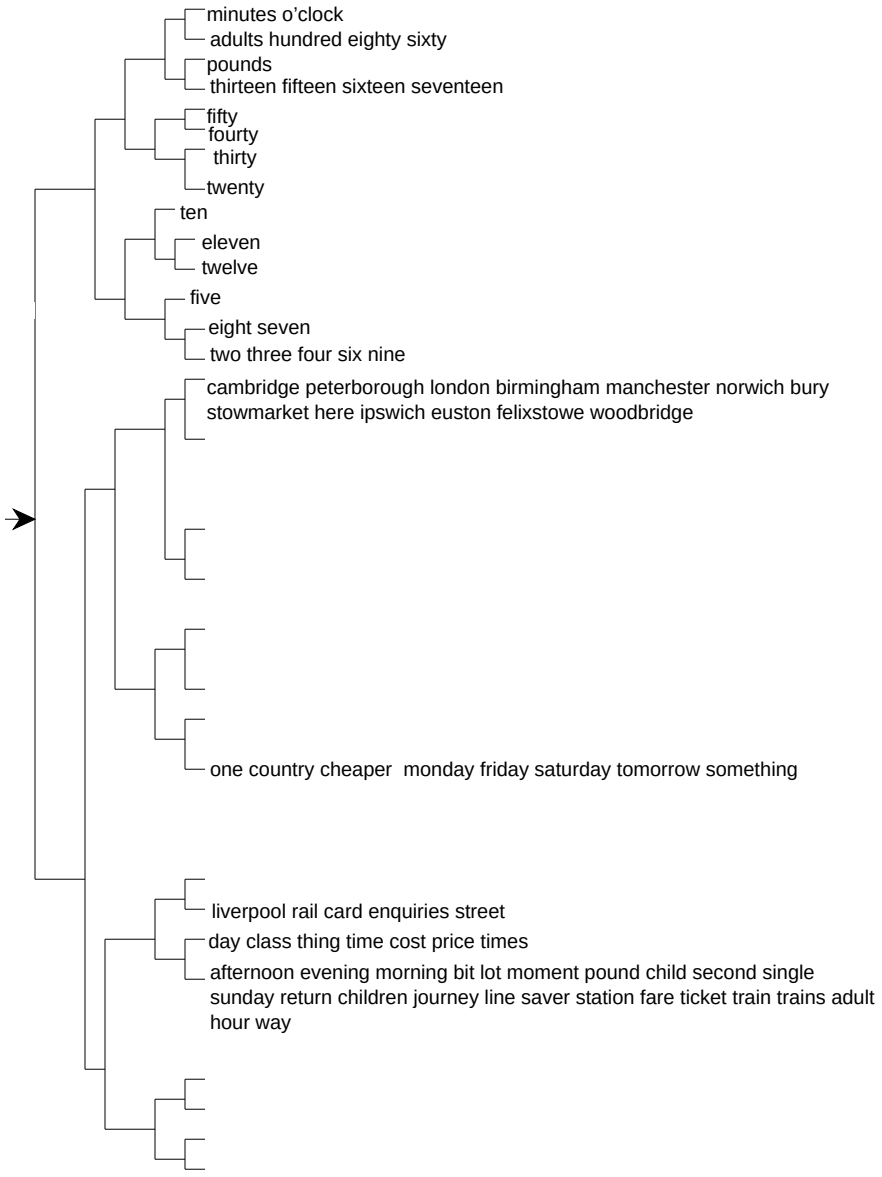


Figure 6.4: Classification of the most frequent nouns of a formatted VODIS corpus, using a merge-based method.

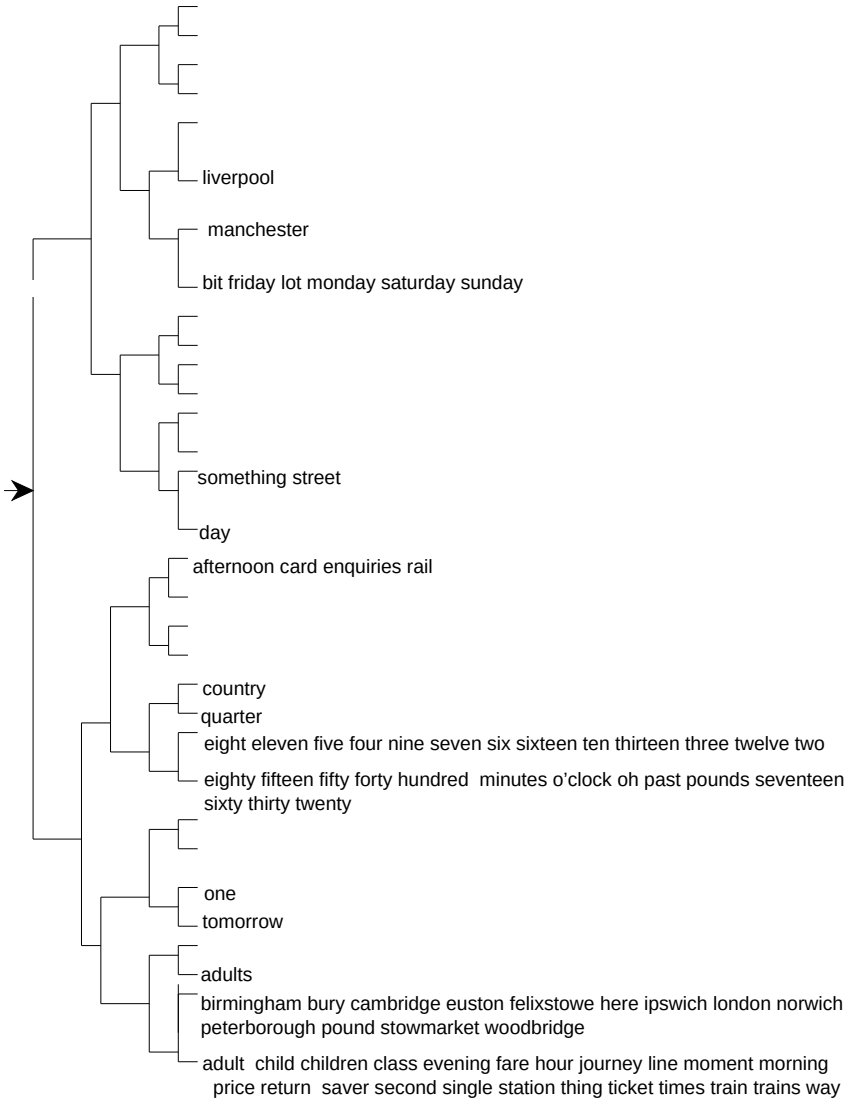


Figure 6.5: Classification of the most frequent nouns of a formatted VODIS corpus, using an annealing-based method.

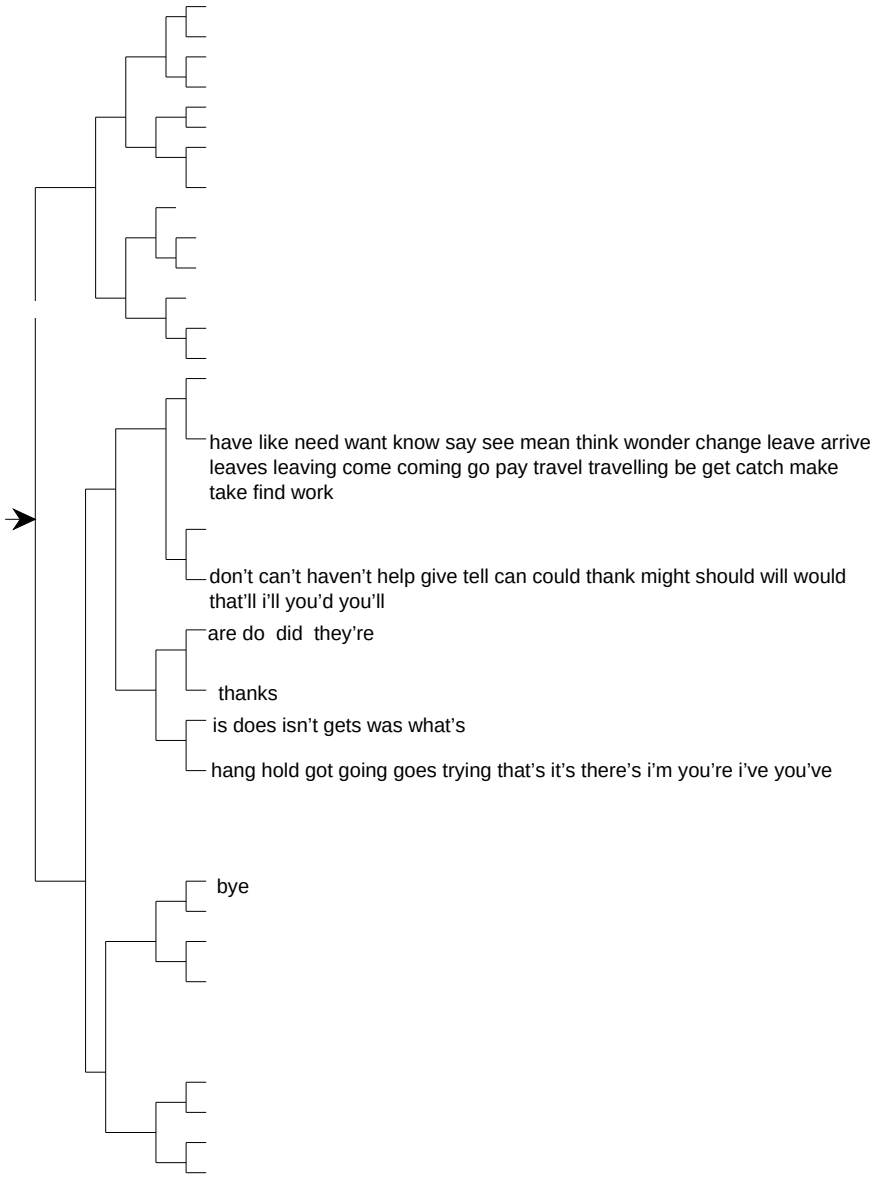


Figure 6.6: Classification of the most frequent verbs of a formatted VODIS corpus, using a merge-based method.

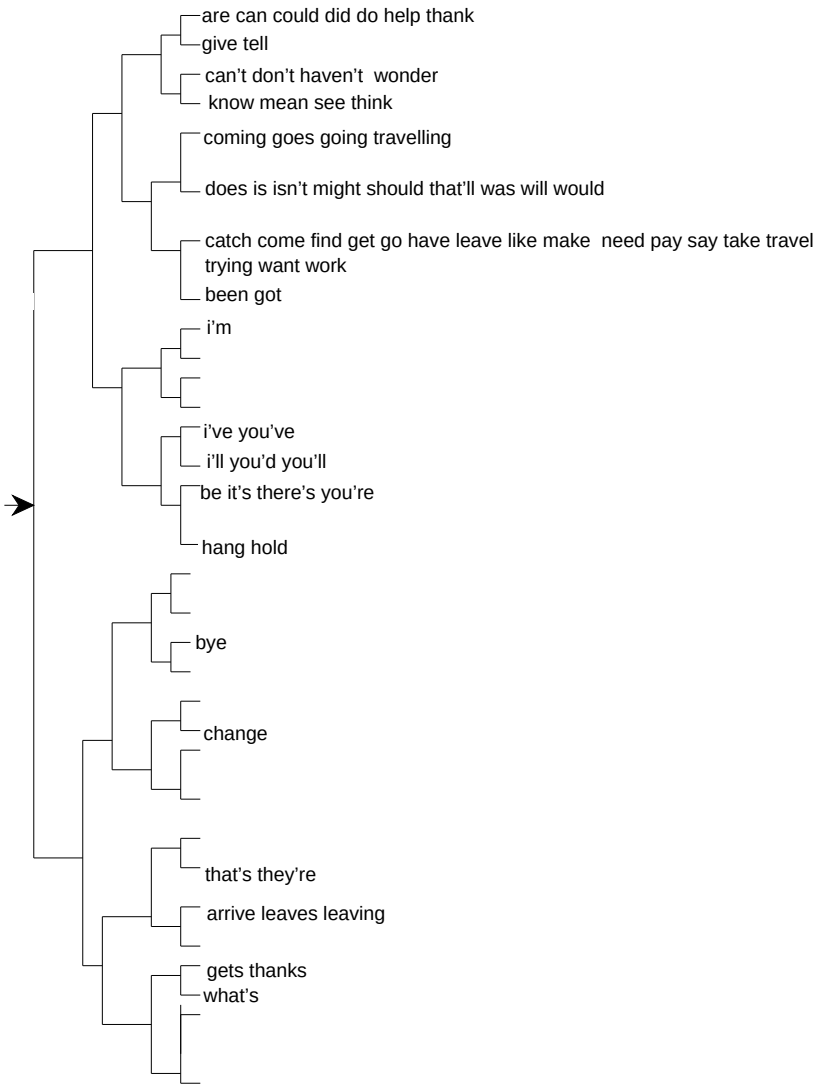


Figure 6.7: Classification of the most frequent verbs of a formatted VODIS corpus, using an annealing-based method.

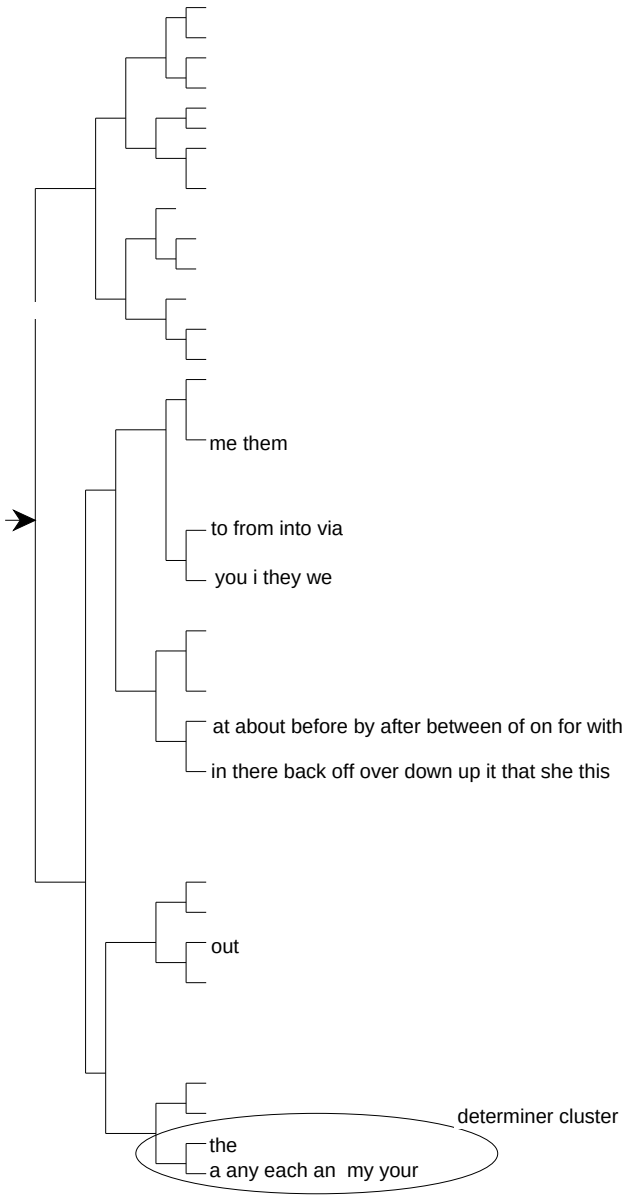


Figure 6.8: The most frequent pronouns, prepositions and determiners of a formatted VODIS corpus, using the merge-based method.

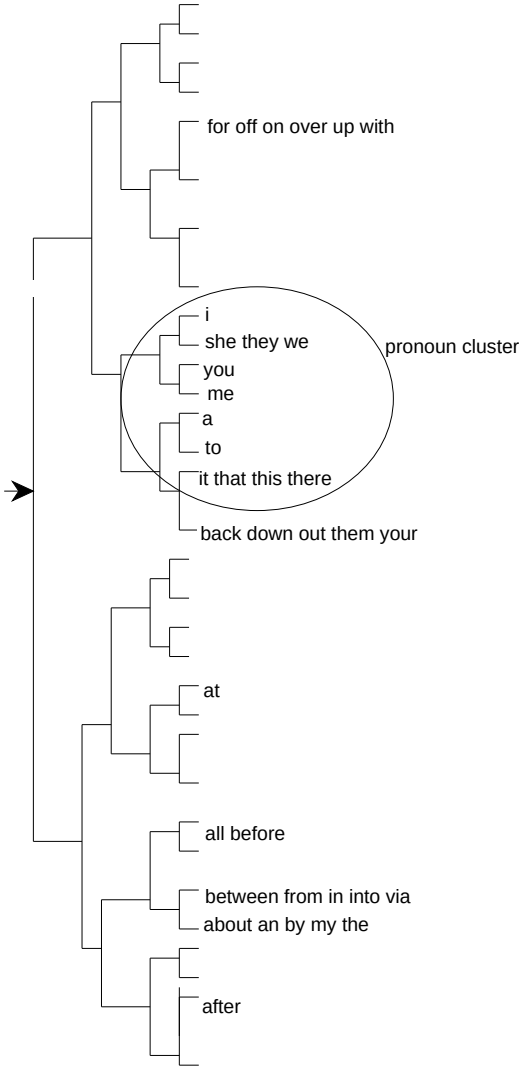


Figure 6.9: The most frequent pronouns, prepositions and determiners of a formatted VODIS corpus, using an annealing-based method.

6.4 Implementing The Hughes-Atwell Cluster Evaluator

As work has progressed in the design of different word classification systems, the need for a quantitative classification evaluator has become more pressing. Recently, Hughes *et al.*. have described a way of testing classification systems of words from the LOB corpus, by comparing word positions in the classification tree with a fixed estimate of the preferred classification for that word.

Classification trees can be sectioned into distinct clusters at different points in the hierarchy (see figure 6.10). Each of these clusters can then be examined by reference to the

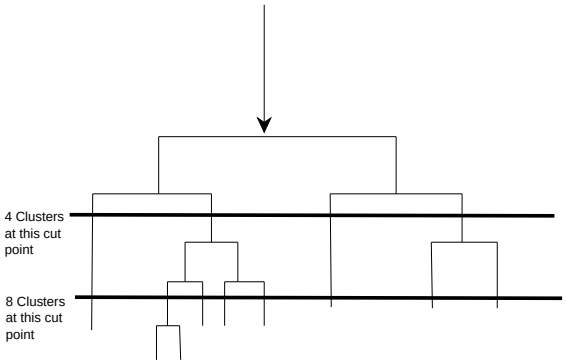
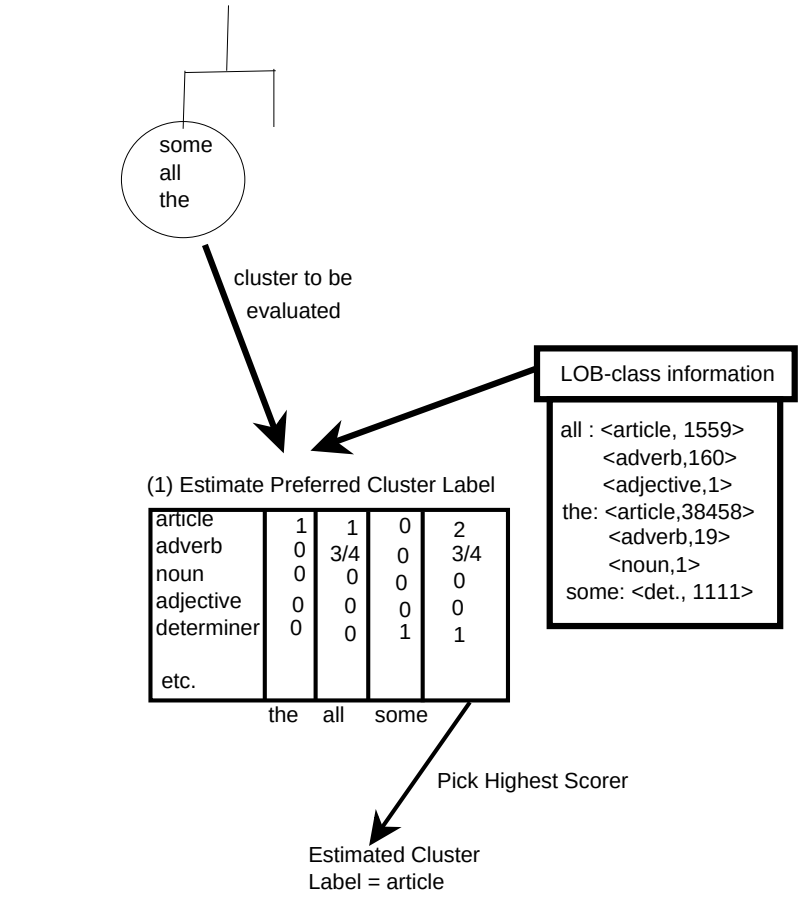


Figure 6.10: Example of how sectioning a cluster tree at different points can produce a different number of separate clusters.

distribution of LOB classes associated to each word member of the cluster. A high-scoring cluster is one whose members are classified similarly in the tagged LOB corpus.

The evaluation is performed on the same 195 most frequent words of the LOB corpus that Hughes uses. The words are automatically classified using the SAMI algorithm. The resulting classification is then passed to the evaluator, which works as follows : the first stage involves producing many successive sections, cutting the tree into distinct clusters, so that an evaluation score can be generated for all possible tree sections; these evaluations can be plotted against number of clusters. At each section, and for each cluster, an estimate of the preferred classification label for that cluster is made. Then, for each member of this cluster, a partial score is calculated which rates the SAMI classification of the word against its distribution of LOB-classes. The summed score is then normalised as a percentage. A broad outline of the evaluation scheme is shown in figure 6.11.

Hughes does not use the classification system provided with the LOB corpus — instead, he uses a reduced classification system which consists of 23 class tags, shown in table 6.1. A C++ class was written to map LOB classes to the new reduced tag set, according to the rules



(2) Update Evaluation Score

(i) "the": matches Estimated Cluster Label with most frequent LOB-class
ADD 1

(ii) "all": also matches Estimated Cluster Label with top LOB-class
ADD 1

(iii) "some": doesn't match Estimated Cluster Label with any of the top four of its LOB-classes
ADD 0

Generally, for each word, if the Estimated Cluster Label matches the kth top LOB-class, ADD (5-k)/4 to the score; k<5. Contributions to the estimate of the Preferred Cluster Label work similarly.

Figure 6.11: Overview of the Hughes-Atwell Cluster Evaluation Process.

ADJ	ADV	ART	CCON
CARD	DET	EX	EXPL
LET	MD	NEG	NOUN
ORD	OTH	PAST	PREP
PRES	PRON	PUNC	QUAL
SCON	TO	WH	

Table 6.1: Reduced tag set used in Hughes-Atwell evaluation system.

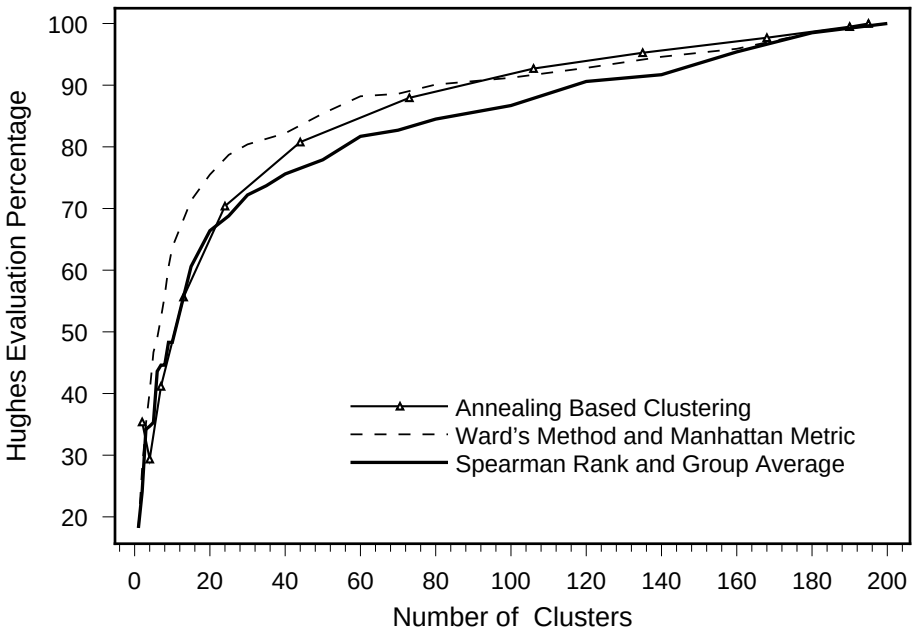


Figure 6.12: Graph showing the performance of the annealing classification system compared to two of the best of the current systems — those of Hughes and Atwell and Finch and Chater. Performance is measured by the Hughes-Atwell cluster evaluation system.

described in Hughes [79]. Another C++ class delivered word classes and frequencies for the words to be used in the evaluation test.

The evaluation was performed on the annealing-based clustering method and a comparison was made with the results presented by Hughes. The results are shown in figure 6.12 [2]. The most important detail about the comparisons which is *not* summarised in this figure is that both of the compared systems use non-contiguous bigrams. Despite this inequality in the raw data being used, the annealing system performs tolerably well. Hughes also gives a single score for his classification system when it is using only contiguous bigrams, at a cut-off point of 25 clusters — the evaluation scores at 76%; the nearest equivalent number of clusters for the annealing results is at 24 clusters — a score of 70.4% is achieved, suggesting that the Hughes method is about 5% stronger at the 25-cluster cut-off point.

The Hughes-Atwell evaluation system is not a perfect measure of the quality of a classification system. Consider figures 6.4 and 6.5 again; the case of the distribution of number words shows that both the merge-based system and the annealing-based system successfully identified this word group, but that the merge-based system does not bind the words together as tightly as the annealing method. We have argued that the annealing system is

[2] The data points for Ward's Method and Spearman's Method can be found in an appendix of John Hughes' PhD Thesis [79]

more successful in this respect. However, if we consider how the Hughes-Atwell evaluator as it calculates the partial contribution of the merge-based number-word clusters, from level 1 onwards, we find that it scores as well as the more well-defined annealing cluster, despite being obviously more fractured. An improved evaluator would take the degree of fragmentation into consideration by possibly weighting each cluster's contribution to the final score by the number of words in that cluster.

6.5 Internal Limitations of the Clustering System

The present algorithm is imperfect in many ways — firstly, it uses a type of simulated annealing which is guided by local optimality considerations only. Therefore the classification which results from such a procedure is unlikely to be globally optimal. No globally optimal solution to the word classification problem has yet to be advanced in the literature. Several immediate improvements could be implemented which would make the optimality less local and more global : more than one word at a time could have its place in the classification altered; the drawback to this improvement is measured in computational expense. The final chapter contains some more suggestions for improving the current classification system.

An analysis of the present algorithm shows that it has complexity $O(V^3)$, when V is the size of the vocabulary to be classified. Therefore the current algorithm's running time will become prohibitive with vocabulary scale-ups which are orders of magnitude bigger than the present one. This is perhaps not a serious problem considering that it currently can handle vocabularies in the range 10^3 — and everyday vocabulary sizes are estimated to be no more than an order of magnitude larger. The algorithm of Brown *et al.* also has complexity $O(V^3)$. Though he does not mention the complexity of his algorithm, Schütze's algorithm is assumed to be the most computationally efficient of all of those discussed in this thesis; this conclusion is based on the size of the vocabulary he can cluster — over 20,000 words. This is an order of magnitude bigger than the clustering experiments of the next largest classification system. In chapter 7, however, a faster clustering algorithm will be introduced as part of a hybrid system which can cluster on the same scale as Schütze's.

Also, using maximum likelihood bigram probability estimates, even with bigrams, makes the system vulnerable to the sparse data problem. For example, if a bigram pair does not occur in a corpus — *e.g.* ⟨`the of`⟩ — then the maximum likelihood probability is set to zero; this means that the mutual information between ⟨`the`⟩ and ⟨`of`⟩ is set to zero, using equation 5.1. A better estimation of the mutual information between these two words should result in a large negative value, capturing the fact that these two highly frequent words very

rarely occur contiguously. This problem should be alleviated by the use of larger and larger corpora, but it could also be helped if the bigram probability estimate was set to a small non-zero value. Finch notes that performance is improved if a small amount of random noise is added to his statistics. This has a negligible effect on segments with healthy frequencies and tends to keep low-frequency words apart. Also, Church *et al.* [30] advise researchers to avoid the 'failure-to-find' fallacy : no evidence for a phenomenon does not mean that one can claim that it does not occur, since it is possible that the data sample is not representative. In practical terms, we cannot be sure that the reason why the bigram ⟨`the of`⟩ does not occur is the same as the reason why the more innocuous ⟨`seventeen bananas`⟩ doesn't — namely due to sparse data. However, these two cases are different, and perhaps negative evidence can be extracted from the former bigram by taking into consideration its relatively healthy composite unigram statistics. The t-statistic, on the other hand, allows us to find out which words are more likely to come after a particular word than after some other word — giving us precisely the negative information which the mutual information statistic cannot.

Dunning [44] describes the limitations involved in making assumptions about the nature of the distribution of words in a sample of text. The assumption of Normality with word probabilities leads to over-estimation of the probabilities of rare but present words; he suggests that a more statistically well-founded language model — that is, one which makes few assumptions about the type of distribution (say, assuming only a Binomial distribution), or one which makes *no* distribution assumptions — should perform better. An interesting statistical question arising from Dunning's observation relates to the *robustness* of particular language model statistics whenever the preconditions for their use are not met. Also of importance is the degree and type of non-normality of distributions of rare words and how this may be modelled in practical language model systems. If the assumptions upon which the current classification system is based are not valid and if the (maximum likelihood) probability estimates are not robust for rare words, then it is unclear how successful the present classifier would be for these words; however, the semantic results described in chapter 5 suggest that some useful structure can be extracted from several thousand words which, though not rare, do occur infrequently. Context words appear in bursts [32], which suggests that language models which implement a notion of recent memory, or current context, should improve performance [95]; this non-normal distributional feature of language has been studied by psycholinguists with various semantic priming and computational linguistics experiments [60, 112]. Church and Gale [31] show how maximum likelihood bigram probability estimates assume that the bigrams are binomially distributed through a sample, and

how even binomial assumptions do not hold for bigram distributions. A similar point is made by Finch [53], who concludes that rank-based statistics should be better at discovering the structure of language.

The binary representation of the structural tag also has drawbacks. The most apparent is the reduction, at every level, of the possibly large number of significant differences to binary distinctions. It is likely that the dimensionality of word classifications at many levels is not binary. This potential limitation of structural tags, however does not make them impractical. Any classification of dimensionality greater than two can be encoded in a binary representation with appropriately altered decision criteria at the binary nodes. This effect is also exploited by Bahl *et al.* [5], who use a statistically based binary decision tree to improve their language model performance.

Another potential criticism linked to such a binary and eventually static classification is that it finds ambiguity impossible to deal with. Resnik [135] articulates the charge forcefully. The argument starts by acknowledging syntactic and semantic ambiguity as an ingrained aspect of language. For example, the word ⟨`bank`⟩ has at least two distinct sets of semantic distributions (corresponding to the two common meanings associated with the word); it also has at least two syntactic distributions — as a noun and a verb. In binary classifications, however, there is only one place where the word can reside. Therefore, it is claimed, the systems cannot fully capture ambiguity, even in principle.

In reply to this criticism, it should be remembered that it is entirely possible that one region of the classification space contains undisputed nouns, another contains undisputed verbs, and another contains that set of nouns which also have verb-like distributions. As long as there are enough statistics to support such classifications, there is no reason why they cannot be made. Indeed, Finch and Chater [50] and Hughes *et.al* report such a classification. Theoretically, if the sum of a word's distributional properties can be expressed using a finite amount of information, then, given enough statistics, an appropriate classification can be derived. If this is not the case — if, for example language displays strong chaotic or fractal properties [137, 118], then such a guarantee cannot be given. Interestingly, the classifications produced by the present algorithm display self-similarity at different classification levels; this phenomenon is also reported in Hughes [79].

Rucker [137] also explains the underlying similarity of digital and analogue information; this is especially pertinent with respect to digital computers, where no numbers are truly analogue. Thus the differences between, for example, the annealing method Pereira and Tishby and the present method are less obvious; however, it is still admitted that, from

a practical perspective, the more analogue-based a classification, the better the results of specific experiments might be. The reason for this is due to the ever-present effects of data-sparseness. There might be an unnecessary duplication of computation which reduces individual class sizes and results in less accurate probability estimates. Resnik [135] makes a similar criticism to Bahl *et al.*'s binary decision tree.

The use of a single most-likely class for words has also been used by Brill *et al.* [15], as part of an automatic word tagging system. They claim that if an automatic part-of-speech tagger uses the simple method of assigning the most probable tag to every word, then 90% accuracy can be achieved.

6.6 Indirect Evaluation

Word classification systems which work automatically are intrinsically interesting; an analysis of their structure and quality is itself an ongoing research topic. However, these systems can also have more immediate uses. The two types of use are related to the two types of approach to the subject — linguistic and engineering. Consequently, indirect evaluation can be linguistic or engineering-based.

Indirect linguistic evaluation examines the utility of the derived classes in solving typical linguistic problems : pronoun reference [45, 55], agreement, word sense disambiguation [101, 59, 162] and resolution of anaphoric reference [22]. A classification is said to be useful if it can contribute to a more accurate linguistic parse of given sentences.

One main engineering application which can use word classes is the statistical language model (see chapter 3). Classifications which, when incorporated into the models, lower the test set perplexity are judged to be useful. The next chapter deals with methods for incorporating classifications into language models, using the structural tag representation.

Chapter 7

Incorporating Word-Cluster Information into Interpolated Language Models

7.1 Overview

This chapter explains a method by which the word classification information which was produced earlier (see chapter 5) can be used in class-based language models (introduced in chapter 3). The preferred language model for many of the present experiments is the interpolated trigram model, where the values of unigram, bigram and trigram weights are calculated by describing the word-probability system as Markovian and using a well-known Markov Model parameter estimation technique, described in section 7.2.

A short survey of research using word classes is given. Following that, in order to compare perplexity values on the same test set, some word based language models are built and evaluated; these experiments also provide a practical introduction to the parameter optimisation method described in section 7.2.

The word clustering algorithm of chapter 5 is computationally intensive; therefore, unfortunately, time limits of this research work prevented the construction of a full-vocabulary cluster using the annealing algorithm. Instead, a less optimal, but quicker supplementary algorithm is introduced in section 7.5. Together with the annealing algorithm, the new hybrid clustering system is introduced and allows greater vocabulary coverage.

Next, experiments involving the class-based language models are described. This chapter finishes with some conclusions of the results of these experiments.

7.2 Markovian Parameter Estimation Using Held-Out Data

Interpolated language models – that is, hybrids of simpler language models — are among the most common in successful language modelling systems [136, 95, 42, 6]. Often, the component language models have complementary strengths and the hybrid is constructed in such a way that maximum weight is given at any one time to the most informative and reliable component. The greater the degree of flexibility in distributing weights across the components, the higher the performance. For example, the simplest interpolated trigram language model has two independent weights, λ_u, λ_b — the trigram weight, $\lambda_t = 1 - (\lambda_u + \lambda_b)$. Given this hybrid, some method needs to be introduced which can optimise the independent weights and hence optimise the performance of the hybrid.

A more complicated hybrid system can be built by making the weights depend on some other, easily calculated parameter. For example, in the estimation of test-set perplexity, the frequency of the previously processed word is readily available. Whereas in the simple hybrid system, each stage of the processing gives fixed weight to each of the three components, in this new hybrid, the weights can vary throughout the test set. For example, if the previously processed word occurs with very low frequency, it might make sense to have a λ_i distribution which minimises the influence of the trigram component — that is, sets λ_t to a small value — and maximises the influence of the less informative, but more reliable bigram and unigram components; conversely with a previously processed word which has high frequency. The problem of selecting a set of weights which are in some way optimal becomes more apparent in this case, since there are usually hundreds of different frequency-dependent weights.

In order to use a parameter estimation technique from Markov modelling theory [129, 121, 122] it is useful to think of the hybrid language model as a Markov chain with two types of arc — emitting and non-emitting. Figure 7.1 shows a single transition for the simple hybrid case. The probability of word w_k following the segment $\langle w_i, w_j \rangle$ is equivalent to the two-stage transition from the state $\langle w_i, w_j \rangle$ to the state $\langle w_j, w_k \rangle$, which is equal to the sum of all ways of making that transition; that is

$$P(w_k) = \lambda_u \times P(w_k) + \lambda_b \times P(w_k|w_j) + \lambda_t \times P(w_k|w_i, w_j)$$

which is a version of the more general equation (2.4) introduced in chapter 2.

Once the language model has been described as Markovian, the weight optimisation problem is re-described as a Markov model parameter estimation problem — in particular, a transition probability estimation problem. The emitting transition probabilities are estimated as usual in a maximum likelihood way, using a training text. The non-emitting, weight proba-

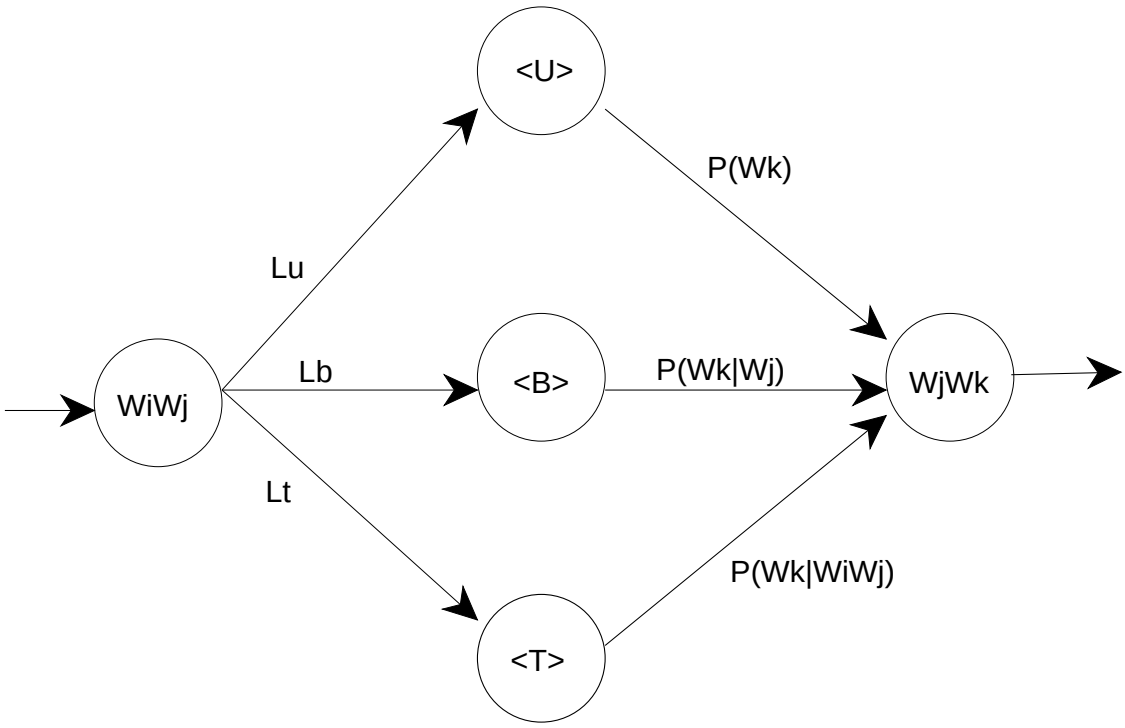


Figure 7.1: Section of a Markov Chain showing the transition from the state corresponding to word-pair w_i,w_j to the state corresponding to word-pair w_j,w_k. The first three arcs, Lu, Lb and Lt correspond to the non-emitting unigram, bigram and trigram transition weights λ_u, λ_b and λ_t. The second set of arcs correspond to the maximum likelihood conditional probabilities of the word w_k, for unigram, bigram and trigram language models.

bilities are estimated using a separate *held out* text.

7.2.1 Transition Probability Estimation

A simplified version of the *Forward-Backward* algorithm [122] can be used iteratively to optimise a set of initial parameter values, to some previously specified degree of significance. The update equation for the jth weight, q_j out of L language models is as follows :

$$q'_j = \sum_{i=1}^{n} \frac{q_j \times P_{LM_j}(w_i)}{\sum_{k=1}^{L} q_k \times P_{LM_k}(w_i)} \tag{7.1}$$

where the held out corpus is n words long and $P_{LM_j}(w_i)$ is the probability estimate of word w_i, using the jth language model.[1] This procedure has been shown to lead to Markovian language models where $P^t(w_1^n) \leq P^{t+1}(w_1^n)$ — *i.e.*, given that the held out text is a sufficiently representative sample of the language being modelled, then the simplified Forward-Backward algorithm makes the held-out text iteratively more likely. The held-out text should be disjoint from the test and training sets to prevent over-learning of those texts.

With the more complicated interpolated language model — where lambda values depend on frequencies, $q'_j(f)$ is calculated using an equation similar to 7.1 above, except that only those words w_i are used which come after a word whose frequency is f.

7.3 Word Classes and Statistical Language Models — A Brief Survey

Rabiner and Huang [129] and Cox [38] provide details of the mathematics behind Markov models.

Most of the recent work on improving the performance of statistical language models by adding in word class information rests on using corpora which have been syntactically tagged. Hull [81], for example, uses a tagged version of the Brown corpus to gain statistics for training a Hidden Markov model, the states of which correspond to syntactic classes. The system he develops is principally an automatic part of speech tagger. Rohlicek, Chow and Roucos [136] decide to diagram sentence fragments using about 100 manually constructed sentence patterns. This reduces the sparse data problem and allows Markov model parameters to be optimised. Their hybrid linguistic-statistical model performs better as a result. Derouault and Merialdo [42] build an interpolated language model based on the word tags of a machine-readable Stenotypy-French dictionary, with some manually adjusted tags at places

[1] Appendix B contains a derivation of this update equation.

where they considered the dictionary tags insufficient. They compared the performance of their system with one based on 200 context-free rules. The systems achieved similar performance levels; the authors also note that the effort invested in creating the 200 context-free rules greatly exceeded that of training Markovian parameters. They also showed that a hybrid grammar-Markov system performed much better than either system by itself.

Han, Park and Choi [74] make the controversial claim that Markov nets with a stack for simulating recursion can properly model natural languages. They also manually analyse several hundred sentences and use this information as training input for their Markov model. Kupiec [96] uses a hidden Markov model to tag words automatically. He designed a network for modelling high-order context manually, by analysing the performance errors of a lower level, fully automatic system.

Srihari and Baltus [150] and Finch [53] both develop *hypertag* systems — where sentence fragments are automatically extracted from a test set. Srihari *et al.* use a tagged corpus, whereas Finch bootstraps his hypertags. The resulting hypertag frequency statistics can be used in a language model.

Jelinek [85] and Brown *et al.* [19] both use automatically extracted word class information to improve languge model performance, as measured by perplexity. They report significant incremental improvements to their models, which currently generate some of the lowest perplexity scores.

7.4 Word Based Interpolated Language Models

The parameter estimation system described above was designed and implemented for lambda values which depend on the frequency of the previous word, and also for lambda values which do not depend on any other value. Also, a language model system was implemented which calculates the test set perplexity of several interpolated language model variants. In each of the experiments described in this chapter, the experimental details are similar unless otherwise specified : a formatted version of the Brown corpus provided all of the data; 10% was used as the test set, 60% was used to gather maximum likelihood probabilities and the remaining 30% was used as the held-out text. The Brown corpus was formatted as follows : it was de-capitalised, de-punctuated (except for the apostrophe) and special characters were removed. The following is an example of the state of the formatted corpus used in these experiments :

```
the fulton county grand jury said friday an investigation of
```

```
atlanta 's recent primary election produced no evidence that
any irregularities took place the jury further said in term
end presentments that the city executive committee which had
over   all charge of the election deserves the praise and thanks
of the city of atlanta for the manner in which the election was
conducted the september   october term jury had been charged by
fulton superior court judge durwood pye to investigate reports
of possible irregularities in the ...
```

Each of the domain types was distributed across the three data sets in accordance with the 10, 60, 30 percentage split.

Perplexities in all cases are estimated according to equation 2.10; as with other systems [121], test set words not present in the training set are ignored and the perplexity estimation is adjusted accordingly. Keeping as many experiment details as possible fixed throughout this series of perplexity estimations allows us to evaluate the performance of the various language models with much confidence.

7.4.1 Component Word Unigram Language Model

The first experiment in this series estimated the test set perplexity of a single word-based unigram language model. This provides us with a pragmatically sensible measure of worst-case performance — an absolute worst-case system is where all training-set words are assigned random probability values, with the constraint that the sum of these probabilities is 1. Another poor-performance system sets all V training-set words as equiprobable : $P(w) = \frac{1}{V}$. In this case, the perplexity, which can be thought of as the average branching factor, is set to V. Since there are 33,360 word types in the training-set, the equiprobable language model has a perplexity of 33,360. The unigram word-based language model is a more pragmatically sensible worst case system because it uses training-set frequencies. The model is described by the following expression:

$$P(w) = \frac{f(w)}{\sum_{i=1}^{V} f(w_i)}$$

where $f(w)$ is the training-set frequency of space delimited word w. The model has a test set perplexity of 1,226.75. The use of a non-equiprobable model represents the biggest single-step improvement in language model performance; after this, it becomes harder to design models which reduce entropy significantly.

7.4.2 Two-Parameter Interpolated Uni-,Bi- and Trigram Language Model

In this experiment, three simple language models are interpolated with two independent lambda values — $\lambda_t = 1 - (\lambda_u + \lambda_b)$ — to give a simple hybrid language model. The system is described mathematically as follows :

$$P(w_k) = \lambda_u \times P(w_k) + \lambda_b \times P(w_k|w_j) + \lambda_t \times P(w_k|w_i, w_j)$$

where w_k is the word to be processed and w_i and w_j are the previous two words processed.

The weights were set initially as $\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$ and the simplified Forward-Backward algorithm was used with a stop condition that $|\lambda^{t+1} - \lambda^t| < 0.001$, for all of the three lambda values. The final $\{\lambda_u, \lambda_b, \lambda_t\}$ values were $\{0.554, 0.415, 0.031\}$ respectively. These values show the extent of the sparse data problem, particularly for trigrams : if the statistical quality of the unigram, bigram and trigram frequencies is good, then the Forward-Backward algorithm would always weigh heavily in favour of the trigram component. The extent to which it doesn't provides a measure of the sparse data problem, for corpora with 10^6 or so words — in this case, only 3% weight is given to the trigram component.

The two-independent parameter interpolated language model results in a perplexity of 701.709.

7.4.3 Frequency Dependent Interpolated Uni-,Bi- and Trigram Language Model

The next language model to be evaluated is the hybrid unigram, bigram and trigram system where lambda parameters depend on the frequency of the previously processed word. In the 33,360 word-type training vocabulary, there are only 428 distinct frequency values, the lowest frequency value being 1 and the highest being 41,346. For each of these 428 frequency values, a unique $\{\lambda_i, i = 1, 2, 3\}$ set is maintained. As in the previous experiment, all initial lambda values are made equiprobable.

After these estimates have been optimised using the simplified Forward-Backward algorithm, a strong frequency-dependent differentiation appears in the λ_i sets. It is interesting to examine the nature of this dependence for any suggestion of pattern which might be exploited in the weight-frequency relationship. Some work has been done in this area already with a frequency-dependent bigram and unigram interpolated language model [122]. Figures 7.2 to 7.4 show a degree of smoothness in high and low frequency areas, suggesting that some approximating curve might be fitted to the graph in these regions with only a small degree of performance deterioration. However the middle frequency range does not seem to fit any

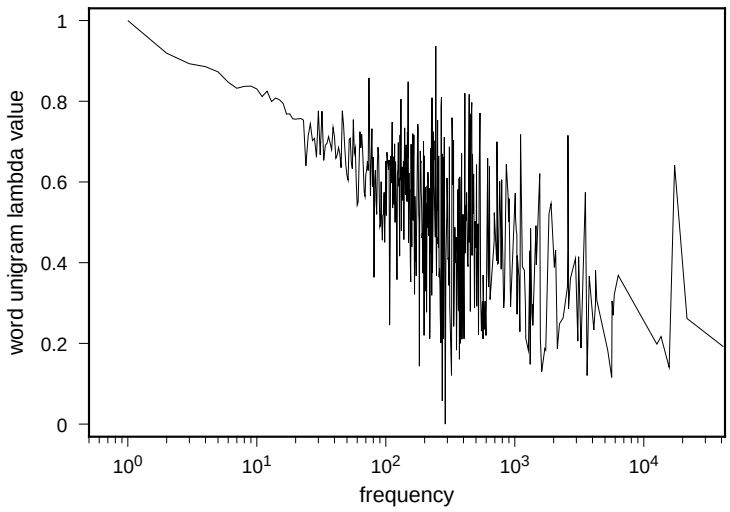


Figure 7.2: Relationship between unigram lambda value and frequency for and interpolated uni-, bi- and trigram word language model. The middle frequency range indicates a noisy weight-frequency relationship.

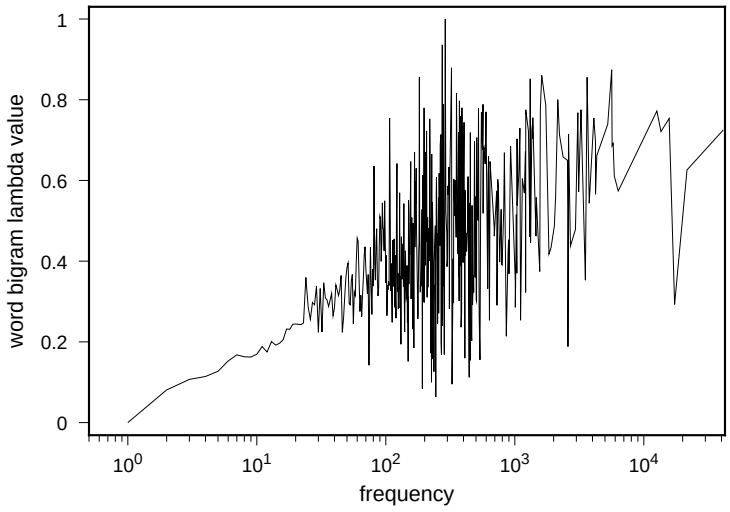


Figure 7.3: Relationship between bigram lambda value and frequency for and interpolated uni-, bi- and trigram word language model. The middle frequency range indicates a noisy weight-frequency relationship.

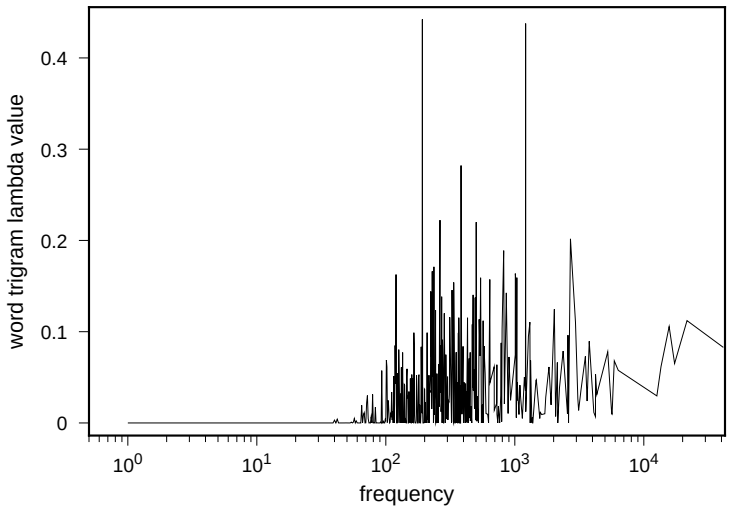


Figure 7.4: Relationship between trigram lambda value and frequency for and interpolated uni-, bi- and trigram word language model. Absence of smoothness in this figure is more likely to be due to sparse data problems.

simple curve — this could be because of the sparseness of data in estimating lambda values for frequencies in this range; if this is so, larger corpora might smooth the curves in these regions; alternatively, the 'apparent' randomness of this region might be inherently random and no amount of data could smooth the data points; in this case, the information content of the best-fit curve for this region would be as large as the information content in the data points themselves — in other words, the data points might be incompressibly random [24]. A third explanation of the nature of the data points in the middle frequency range is that they exhibit low order chaos. There are techniques available [152] for differentiating low order chaos from genuine randomness, and also there is a movement within mathematical approaches to language phenomena which incorporates the techniques of chaos theory and information processing [118, 117].

The low range frequency values have many exemplars — that is, there are many words in the training corpus whose frequency is equal to some low value f_l; conversely, the high value frequencies, f_h correspond to single words which occur many times. This phenomenon of stability in low frequency yet frequent and in high frequency yet unique words is mirrored in some work on the reliability of semantic properties of words; Liddy and Paik [101] report that both unique semantic descriptions of words and high frequency semantic descriptions of words (where the semantics comes from a machine-readable dictionary) provide the most reliable predictive indicators for the task of word sense disambiguation.

The optimised parameters are fitted into an interpolated language model the core of which

is described by the following expression :

$$P(w_k) = \lambda_u(f) \times P(w_k) + \lambda_b(f) \times P(w_k|w_j) + \lambda_t(f) \times P(w_k|w_i, w_j)$$

where $f = f(w_j)$ if there is a valid w_j and 0 otherwise — for example, at the beginning of the test set, and when the previous word is not in the training vocabulary. The λ_i values for frequencies of 0 are $\{1.0, 0.0, 0.0\}$, since no context is known and these values are equivalent to using the unigram word model only (assuming, of course, that the unigram model is the first component of the hybrid system).

This language model reduces the perplexity to 621.649. The extra expense of allowing the lambda set to vary across previous-word frequency has usefully reduced perplexity when parameters are optimised using the empirical Forward-Backward algorithm.

7.4.4 Weighted Average Language Model

It was suggested in the previous section that curve-approximation could be used to replace the computationally expensive and data-intensive Markov parameter optimisation technique. A successful model of this type was designed by O'Boyle [119], which has the extra advantage of being able to use segments longer than three words if they occur with sufficient statistical significance. This model is described as follows :

$$P(w_k|w_1^{k-1}) = \frac{\sum_{i=1}^{m} \lambda_i \times P_{ML}(w_k|w_{k-i}^{k-1}) + \lambda_0 \times P_{ML}(w_k)}{\sum_{i=0}^{m} \lambda_i}$$

where there are significant segments up to $m+1$ words long and $P_{ML}(w_k)$ is the maximum likelihood probability estimate of a word. The numerator acts as a normaliser. It has been found that

$$\lambda_i = 2^{(|w_{k-i}^{k-1}|)} \times \log f(w_{k-i}^{k-1})$$

where $|w_{k-i}^{k-1}|$ is the size of the segment, results in a useful language model.

This model was implemented and the resultant test set perplexity was 630.901. Thus its performance is comparable to the most successful of the interpolated systems, though it is slightly worse. However, the two experiments do not use the same amount of data. The weighted average system does not need any parameter training, so the held-out corpus was not used in the above experiment. O'Boyle has shown that using all of the held-out data results in a model whose perplexity is approximately equal to that of the frequency-dependent interpolated trigram model.

7.5 A Less Optimal Word Clustering Algorithm

The previous word-only perplexity experiments provide a relevant and varied context within which class-based language model performance can be evaluated. But before these experiments which use class information can be performed, some more work needs to be done to arrive at a classification which includes the entire 33,360 word training vocabulary and which takes a reasonable amount of time.

To date, no automatic word classification system has clustered this amount of words — the largest coverage is with a system that is described by Schütze [143]. The annealing algorithm described in chapter 5 takes a vocabulary portion and allows the sth bit of any word to be flipped, at any time during the sth processing stage. This produces successful clusters in a respectable time for small vocabulary subsets of between 500 and 1,000 words; however, the algorithm has complexity $O(V^3)$, which means that the two orders of magnitude required to move from a 300-word vocabulary to a 30,000-word one results in a prohibitive running time. One top-down strategy for classifying low frequency words involves using the last three letters of the word as a clue to its most likely classification [15]. In effect, a relative entropy calculation finds the most likely class given the three letters. This gives 79.5% accuracy for English. That method is not implemented here; instead a less accurate bottom-up method is used.

A hybrid system can be used to produce mixed quality large-vocabulary word classifications. First, the most frequent vocabulary words — say, the first 500 or so — are passed to the computationally intensive annealing algorithm, which gives a high quality classification structure for only these words. Let these words belong to set V_a. Next, this clustered word set is 'grown', word by word, starting with the most frequent vocabulary item *not* already processed by the annealing algorithm; that is, the most frequent word from the set $V - V_a$. Instead of flipping the sth bit of every word to find the best bit values for the entire growing vocabulary, the original set of words V_a are never altered again; only the bits of the single added word are flipped to find its ideal position, given a fixed V_a classification. This means that only 16 decisions are needed to fully classify any new word (although each decision involves the calculation of two average class mutual information scores).

The justification behind this hybrid system is as follows : Zipf's Law [164] ensures that the most frequent words account for the majority of word tokens (for example, the most frequent 1,000 word types account for 85% of word tokens); these words strongly influence the overall shape of the classification; also, the frequency statistics associated with their unigrams and bigrams are reliable; the less frequent a word, the less successfully it will be classified and

the less effect it will have on the overall structure of the classification. This exploitation of a regularity of language, discovered by Zipf, mitigates the assumption that a new, relatively low frequency word will not have *any* significant effect on the classification of more frequent words; hence we do not need to re-classify high frequency words each time new, low frequency ones are added.

Unfortunately, the machine which runs these experiments cannot contain the unigram and bigram frequency information for all 33,360 words in RAM. The largest vocabulary size it can handle is about 15,000 words. This isn't too limiting a problem, since the 15,000th most frequent word in the training vocabulary only has a frequency of 2. Such a low frequency word will not be clustered successfully by any algorithm. Therefore, only the first 15,000 words need be clustered by the fast cluster algorithm; the remaining words can be assigned random unique classification positions.

7.6 Class Based Interpolated Language Models

A classification of the training vocabulary was produced, by the hybrid method described in the last section. First, the 633 most frequent words of the training set taken from the Brown corpus were classified using the SAMI algorithm. Then, the next most frequent 14,367 words were clustered, using the fast algorithm; these words were processed from most to least frequent. Then, the remaining 18,360 words were added to the classification by assigning unique random TAG values to them. Databases containing unigram, bigram and trigram frequency information were constructed for each of sixteen bit-levels. These databases are CC-databases (see chapter 3).

7.6.1 Two-Level Interpolated Models

This section reports on a series of interpolated language models the components of which are drawn from two levels of the classification structure. One set of models is always drawn from level 16 — they corresponds to word-based language models; the other set is drawn from one of the other 15 levels.

It should be noted that unigram language models cannot be improved by using class information — that is, unigram models at all levels lead to exactly the same perplexity scores and in conjunction with other language models, contribute in similar ways. This is because

$$P(C) \times P(w|C)$$

reduces to

$$P(w)$$

no matter what classification system is used, for systems which have one tag per word. This information reduces the number of language model permutations, and also reduces the number of independent parameters to be optimised. For example, if we wanted to see how well a 16-12 interpolated trigram language model performs — that is, a hybrid model whose components are taken from level 16 (words) and level 12 — we only need interpolate the five language models P_u^{16} (word unigram model), P_b^{16} (word bigram model), P_t^{16} (word trigram model), P_u^{12} (s=12 unigram model) and P_b^{12} ($s = 12$ bigram model). The $s = 12$ trigram model, P_t^{12} is functionally equivalent to the word unigram model. Formally, each of these two-level systems can be described as follows :

$$\begin{aligned} P(w_k) = & \\ & \lambda_u^{16}(f) \times P_u^{16}(w_k) + \lambda_b^{16}(f) \times P_b^{16}(w_k) + \lambda_t^{16}(f) \times P_t^{16}(w_k) + \\ & \lambda_b^{i}(f) \times P_b^{i}(w_k) + \lambda_t^{i}(f) \times P_t^{i}(w_k) \end{aligned}$$

where P^i corresponds to a class based probability estimator which uses the ith classification level and $f = f(w_{k-1})$ if such a word frequency is defined, and 0 otherwise.

Fifteen independent experiments were conducted on two-level interpolated trigram language models. In every case, the extra class information does improve the hybrid model. Obviously, some classification levels are more successful than others in the quantity of the improvement they offer — for example, an $s = 15$ classification is very similar to an ordinary word-based language model; an $s = 1$ classification may be reliable, but the extra information imparted by class knowledge is minimal. Figure 7.5 summarises the test set perplexity results of these experiments. The best language model is at a bit depth of $s = 8$: here, the test set perplexity is 586.923, compared to a standard word-based test set perplexity of 621.649. This is a 5.6% improvement in language model quality.

7.6.2 Parameter Convergence Criteria

A slightly confusing feature of figure 7.5 is that at certain points — for example, from $s = 11$ to $s = 12$ — there is an unexpected dip in the test set perplexity score. This feature is mirrored in the transition from $s = 4$ to $s = 3$. In order to explain this, we decided to investigate how Markovian parameter convergence criteria influenced the final test set perplexity of hybrid model.

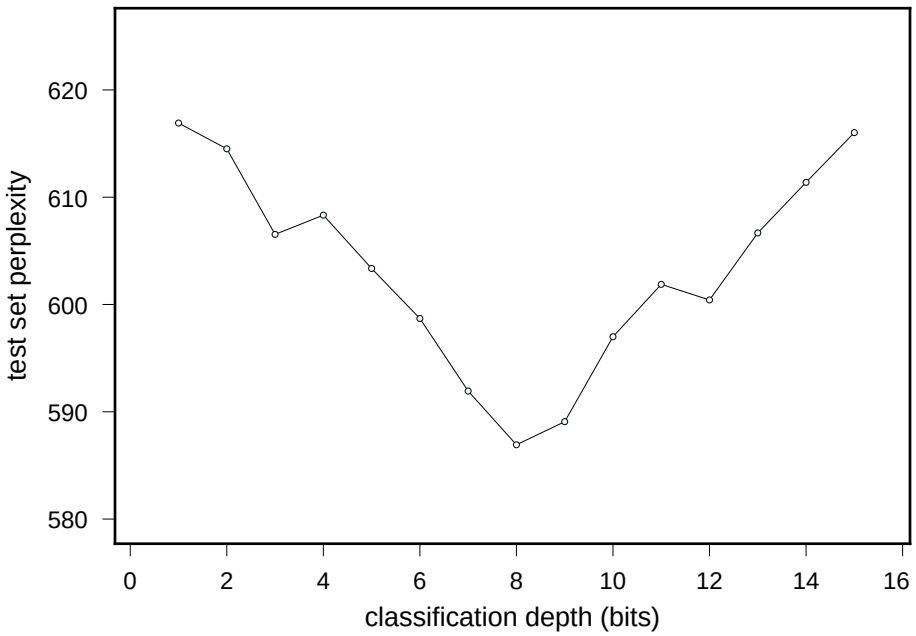


Figure 7.5: Test set perplexity results for fifteen two-level hybrid language models, showing significant perplexity reduction, with the greatest effect at $s = 8$. For comparison, a standard word-based language model scores 621.649.

We concentrate on the three hybrid models corresponding to bit depths $s = 10$, $s = 11$ and $s = 12$ and are especially interested in the number of iterations it takes for each of these three systems to meet our convergence criterion, $|\lambda^{t+1} - \lambda^t| \leq 0.001$. To this end, we calculated the test set perplexity of these models at various stages and plotted perplexity against iteration for each model, up to the point where our arbitrary convergence criterion is met. Figure 7.6 shows this graph. While the parameters took 200 iterations to converge in the case of $s = 12$, for $s = 11$ and $s = 10$, they took 96 and 107 iterations respectively. We noticed that the pattern of convergence in these three cases seemed similar. Whilst the convergence criterion is arbitrary for the purposes of comparing these language models, we concluded that making the degree of significance an order of magnitude smaller — from 0.001 to 0.0001 — would lead to more settled results. To test this hypothesis, we allowed the $s = 10$ model to converge beyond 0.001. As we can see in figure 7.7, showing the test set perplexity results for $s = 10$ beyond 0.001, the convergence point is lower and more stable; also, it approximately follows the same convergence pattern seen in the $s = 12$ system.

7.6.3 Three-Level Interpolated Models

Ideally, the class-based language models described in this thesis which use structural tags will perform best whenever all sixteen (generally, n) levels are available simultaneously. Un-

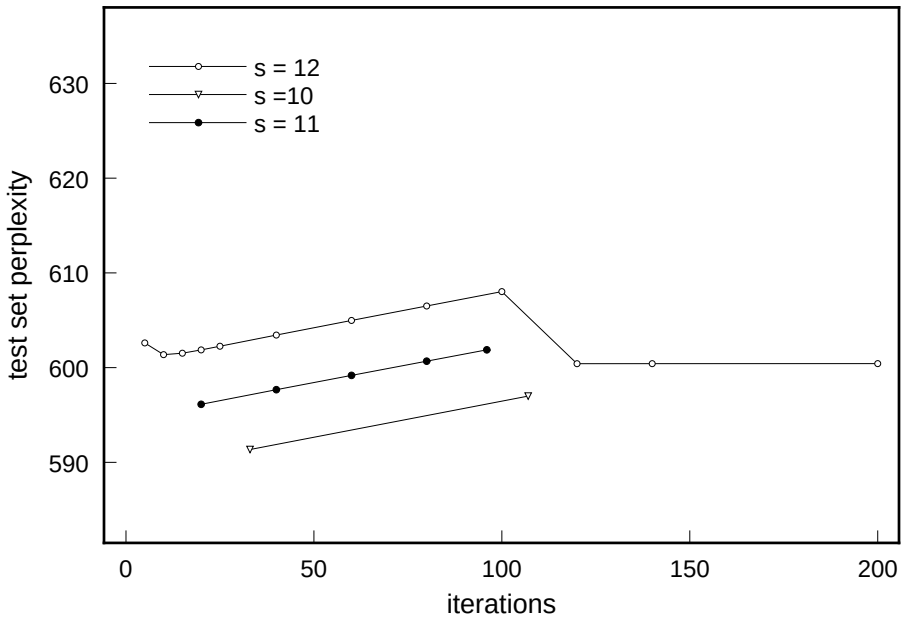


Figure 7.6: Perplexity against iteration stage for two-level s=10, s=11 and s=12. The convergence pattern is similar in all three cases, though the models have not yet finally converged.

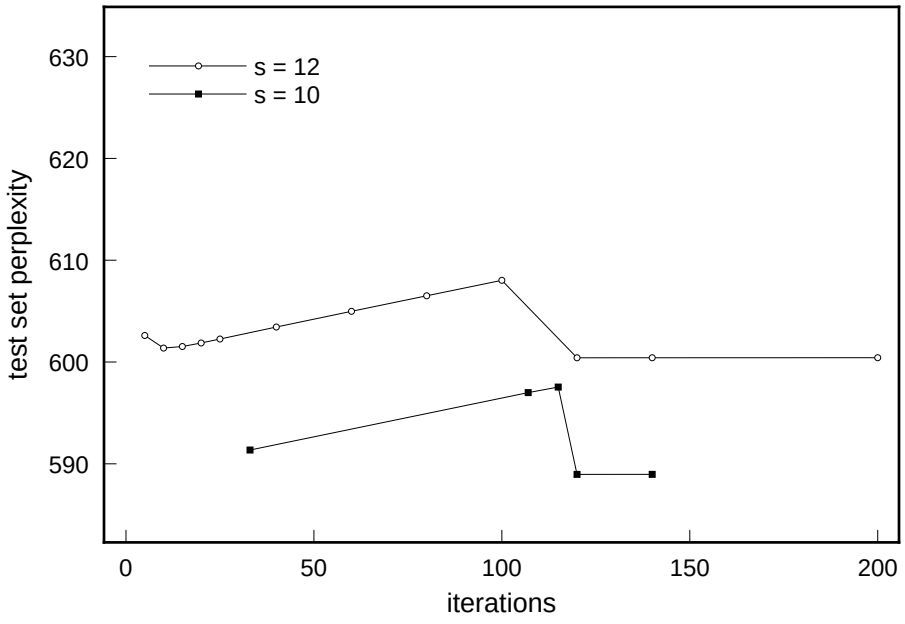


Figure 7.7: Perplexity against iteration stage, for two-level s=10 and s=12, beyond convergence criterion 0.001; the convergence is more stable.

fortunately, the more models there are, the more training data the Markovian parameters require, and the current experiments are based on a corpus which is only one million words long, including training data, test set and held-out data.

Already with a two-level interpolated model, there are in total 2,140 parameters — 1,712 independent parameters — to be optimised using a 300,000 word held-out text. Each new level added to the interpolated system requires approximately 850 new λ-parameters to be trained. We performed one three-level experiment — using the 16-8-5 model and found that the perplexity was only slightly better than the best two-level system, which was the 16-8 model. We expected that the improvement would be slight, since the new three-level model incorporates the best two-level model, and also because of the increase in the number of parameters which need to be trained. The three-level model produced a test-set perplexity score of 586.543 — 0.38 of a perplexity point better.

7.7 Conclusion

The automatically generated class information produced by using the SAMI algorithm does improve the performance of statistical language models; the simplest system — a two-level one — makes a 5.6% perplexity improvement. Experiments using all sixteen levels require larger corpora, though we suggest that these will perform better that two-level systems. Results from a single three-level system lend some support to this belief. We ended the previous chapter by claiming that one useful way of evaluating the utility of a classification system was by doing so indirectly. Results from this chapter, then, are offered as further evidence of the utility of our classification system. Time limits and corpora size have prevented us from building many-level class-based language models.

The significance of percentage improvements in perplexity is hard to gague — a three percent improvement from a baseline perplexity score of 50 might correspond to a greater achievement than a three percent improvement over a baseline score of 100, for example. For completeness, however, we report that the automatic classification system of Brown *et al.*, when incorporated into a word and class interpolated model, reduced the perplexity from 244 (word-only) to 236 (word and class models), corresponding to a 3.3% improvement. Our 5.6% improvement represents a reduction in perplexity from 621.6 to 586.9, the differences in absolute perplexity values between our system and that of Brown *et al.* being explained partly by the disparity in training set sizes.

Chapter 8

Conclusions

8.1 Overview

In this chapter, we summarise the main results of experiments described in this thesis. We then make several suggestions for further work. Finally, we re-iterate our position on the cognitive relevance of models of this type, concluding that even those who are not interested in cognitive models per se and who would rather build workable engineering applications must eventually confront cognitive issues before they can construct easy to use natural language interfaces.

8.2 Summary of Results

In this section, we briefly recapitulate the experimental results of the thesis. First, we implemented some word-based language models and carried out preliminary experiments to demonstrate the most promising types of class-based language models. Then we turned our attention to automatic word classification.

We successfully designed and implemented the SAMI algorithm to discover some important syntactic and semantic properties of a finite-state grammar described by Elman. Next, we showed that the system could easily scale up to a small sized transcribed spoken English corpus; here, it also discovered syntactically coherent word classes, and some semantic classes also. We then moved to a medium-sized written English corpus and the syntactic and some semantic detail became more pronounced. We also showed that phonemic and, to a lesser extent, letter-based orthographic similarities could be discovered. An experiment which made use of a pre-tagged corpus showed that much more semantic detail resides in the bigram than one might intuitively imagine.

We also supported a hypothesis that our system was not language-dependent by demonstrating syntactic pattern discovery in an ancient Latin corpus. Having exhibited substantive success, we then subjected our algorithm to several evaluation experiments : we showed that it and a merge-based system have complementary strengths and weaknesses. Also, we implemented a benchmark evaluator and demonstrated that our system performs tolerably well compared to others, even though it uses contiguous bigrams only.

Then we returned to language models and designed a series of experiments to show how class information, automatically generated, improves language model performance. We looked at all two-level class-based language models and noted that the best of these systems could improve test set perplexity by 5.6%. To build these language models, we had to construct a secondary word classification algorithm which can deal comfortably with large vocabularies.

Finally, as an illustration of the sort of advantage which the new class-based language models can give us, we return to the oronym introduced in chapter 2, based on the uttered sentence

```
the boys eat the sandwiches
```

Figure 8.1 reproduces these results, along with the probability scores for the same nine oronyms when given as input to the best of the language models developed in this thesis — the 16-8-5 three-level hybrid. The new language model successfully identifies the most likely utterance. Also, the third most likely sentence — ⟨`the buoys eat the sandwiches`⟩ — is now also grammatically well-formed. In all but two cases, sentences which are well-formed receive a higher probability, and vice-versa for ungrammatical sentences.

8.3 Suggestions for Further Work

Many improvements can be made, both to the automatic classification system and to the class-based language models into which we put automatic word class information. Also, we accept that the finitary models investigated in this thesis are inadequate as cognitive grammar models. However, some techniques can be carried across from work on finitary models into more powerful grammar formalisms. Ultimately, we suggest that successful models of language must be models of language *use*.

sentence	W.A. probability $\times 10^{-20}$	16-8-5 probability $\times 10^{-20}$
the boy seat the sandwiches	3418.88	7848.38
the boys eat the sandwiches	1787	8821.03
the boy seat this and which is	435.392	136.652
the boys eat this and which is	231.73	149.16
the buoys eat the sandwiches	194.805	469.113
the buoys eat this and which is	25.3	7.93884
the boys eat the sand which is	13.7556	20.799
the buoys eat the sand which is	1.499	1.10612
the buoy seat this and which is	0	0

Figure 8.1: Nine versions of a phonemically identical oronym, ordered by weighted average (W.A.) probability. The W.A. language model ranks the preferred sentence above all but one of the less favoured options. The 16-8-5 Class-based interpolated model successfully predicts the original utterance as the most likely.

8.3.1 Automatic Word Classification

Larger Corpora

As ever, improvements could be made to the quality of automatic word classification by using bigger corpora. This move brings with it many implementation and hardware pressures — the tools which generate frequency statistics will need to be altered to accommodate corpora of several orders of magnitude bigger; the classification member functions and data structures will need to be adjusted likewise. Storage and retrieval functions might need to be optimised, requiring some major changes in the associated algorithms. All of the software engineering challenges involved in enlarging an existing system will need to be addressed.

Hybrid Systems

In section 6.3.2, we saw how a merge-based system and the simulated annealing approach had complementary strengths and weaknesses. If these two systems could be incorporated into a hybrid system, then each system could be used only where it performs well; the resulting hybrid might result in more coherent classifications. Combining these systems is not easy, though here we offer one suggestion. When we consider the effect of processing on a single word tag/class for each system, we can see that the annealing system reduces the uncertainly of bits from most significant to least significant. That is, at $s = 1$, the first bit of the word

changes from a '#' ('1' or '0') symbol to either a '1' or a '0'. This represents a reduction in uncertainty of one bit. The merge system works from the least significant bit up, but we still do not know where in the overall classification hierarchy the word will settle. If we think of the the annealing process as operating on V groups, instead of V words, then we can see that a word-flip now becomes a group-flip. If we start by calling the merge system, after n iterations, we are left with $V - n$ groups. We can then anneal these $V - n$ groups. The resulting system should, in theory, take less time to run, since it is making fewer computations.

Instead of a one-merge, one-annealing pattern, we could make the merge dependent on annealing by insisting that merging can only take place within some sub-branch of the hierarchy.

Dynamic Classifications

The classification system we developed is static and represents a global word classification approximation. A dynamic classification system could be developed which, at any instant, represents a hypothesis about the domain or topic of interest underlying the stream of words being processed. For example, the TAG value for the word ⟨`bank`⟩ might be in a region of space close to ⟨`money`⟩, ⟨`finance`⟩, ⟨`robbery`⟩, *etc.* if the system hypothesises that the domain or topic is about finance-related subjects. If, on the other hand, the system estimates that the domain is one concerning fishing, then the word ⟨`bank`⟩ might have a TAG which is close to words like ⟨`fishing`⟩, ⟨`river`⟩, *etc.*

Alternatively, there may be many individual classification hierarchies, all fixed, and each one used only if certain domain hypotheses are made. Structural tags can even include words which are classed according to their phonemic similarity, or by any other criteria considered useful to computational linguists.

Multiple Word Tags

One partial solution to the problem of ambiguity is to allow more than one TAG value for each word in the vocabulary. Words which, while orthographically identical, have distinct semantic and even syntactic distributions can then reside in separate parts of the classification space. If many of the TAG versions of some word end up in a small region of the space, then we can conclude that there are too many TAG versions for that word's distributional patterns. Conversely, some way might be found to identify that there are too few distinct TAG versions for a word and so give it a few more.

Improving Mutual Information Estimation

We mentioned that there are two reasons why bigrams do not appear in a corpus : first, they might be absent because of the limited size of the corpus; second, they might be absent because they genuinely do not appear together often in the underlying language. This was seen most clearly in the bigram ⟨`the of`⟩ — the individual unigrams are both of a high frequency, yet the bigram does not appear. The current implementation makes the safe assumption that when no information exists about a bigram, we have to assume a mutual information score of zero. This makes sense because we assume that the vast majority of bigrams do not occur due to sparse data problems.

However, we also note that those bigrams which, by virtue of their high component unigram frequencies, might lead us to expect a non-zero bigram frequency, can be distinguished precisely due to their high unigram frequencies. We suggest, therefore, that a better classification system would result if we were able to assign high negative mutual information values to these pairs.

Another, more attractive alternative is to improve the underlying bigram probability estimates so that no bigram probability is ever zero. This would naturally result in low negative values for the ⟨`the of`⟩ bigram but would estimate near-zero mutual information values for small probability bigrams whose component unigrams also have small probabilities. One such model involves using Turing-Good unigram estimates.

Classifying Low Frequency Words

Brill and Marcus [15] have reported that taking the last three letters of a low frequency word as its representation leads to better classifications for these words. This would result in a great improvement to our large vocabulary hybrid classification system, under half of whose words are still classified randomly. However, this advantage might not hold for all languages.

Incremental Classification

Following from results described by Elman [47] and Carroll and Charniak [23] we conclude that word classification systems which classify words from incrementally more complex grammars will result in better final classifications. Redington *et al.* [132] have carried out work exploring possible relations between this idea and theories of syntactic category acquisition in children.

Exploiting Non-Contiguous Bigram Information

It should be possible to use non-contiguous bigram probability estimates as extra data for the simple bigram mutual information maximising algorithm; this could improve classification performance.

Generalised Mutual Information

It should also be possible to extent the scope of the mutual information classifier to include trigram and generally n-gram information. If sufficient n-gram class information exists, implementing a generalised mutual information estimator should lead to an improvement in the quality of the resulting classifications.

8.3.2 Class Based Language Models

More Training Data

Obviously, class-based language models would benefit from an increase in the amount of training data; the same technical problems associated with scale-up are applicable here.

Weighted Average Class Based Language Models

A hierarchical classification system provides more reliable n-gram statistics as the classification granularity increases. For example, there are only 8 possible class trigram types at classification level $s = 1$, all of which occur a non-zero amount of times. Similarly for the 64 $s = 2$ classification types. This advantage could be exploited in a class-based language model which was not limited to trigrams, but which used word segments as long as they occurred with sufficient frequency. O'Boyle [119] has developed just such a model for words, called the weighted average language model. We suggest that such a model, which has already been shown to perform as well as an interpolated frequency-dependent language model, could best exploit the longer n-gram class information which the structural tag system can offer.

Altering the Dependencies of Markovian Training Parameters

Currently, the lambda parameters which determine how much weight each pure language model is given depend on the frequency of the previously processed word. This gives us 428 frequency bins, for the experiments described in the previous chapter. If we want to use more language models, we begin to find that there are too many parameters for the training data. We could, however, make these lambda values depend on other features of the previous word :

for example, we could make them depend on the frequency of the class (to whatever depth is deemed appropriate) of that word. In other words, instead of making lambda depend on $f(C^{16}(w_{i-1}))$, we can make it depend on $f(C^s(w_{i-1}))$, $1 \leq s \leq 15$. Alternatively, we can make the parameters depend on the classes themselves, instead of their frequencies : $\lambda \propto C^s(w_{i-1})$. We could also make lambda depend on the previous class bigram, at a sufficiently high classification depth so that each class bigram is adequately represented — that is, $\lambda \propto \langle C^s(w_{i-2}), C^s(w_{i-1})\rangle$. In this way, we are using a more linguistic criterion for determining the reliability of the following word.

Setting Initial Weight Values

We normally start each experiment with all m language models equiprobable, at $\frac{1}{m}$, but we are not limited to this uninformative starting point. For the two-level (five language model) trigram interpolated experiments of the last chapter, we set all weights to $\frac{1}{5}$, but we could have tried alternatives; this might lead to faster coverage, or perhaps a better convergence for a given stop condition. In all the two-level experiments, the first three models were just the interpolated unigram, bigram and trigram ones. We already have a final weight distribution for this system. Therefore we could have an alternative initial weight distribution of $\{w_u, w_b, w_t, 0.0, 0.0\}$, where w_u, *etc.* are the final unigram, bigram and trigram weights from the standard word interpolated experiment.

Testing Against a Pre-Tagged Corpus

We have shown that class information, derived automatically, makes language models which use this information more useful. Since we take as our data starting-point the same as a word-based language modeller — that is, corpora with words only — then we have considered it fair to compare these two systems. However, a more challenging comparison is between class-based language models which make use of tagged corpora. We imagine that such models will perform better than our system, but we anticipate that the performance difference, measured by perplexity, might not be too great. The closer the performance of these two systems, the more preferable our fully automatic one becomes — since it has used less data. There are several technical difficulties associated with fair comparison of two systems like these. Similar problems arose when we tried, in the last chapter, to compare the weighted average model with a word-based interpolated model.

Full Defocusing

Time and data restrictions both prevented the development of the defocused class-based model first introduced in chapter 3. We strongly believe that a model which can have access to all sixteen (generally, all s) levels of classification simultaneously will be more useful than one which only allows access to a sub-set of these. We are partially supported in these beliefs by some of the results of the previous chapter.

Also, it is possible to develop a defocused class-based language model without resort to parameter training — using a king of backoff technique. This model might also be worth investigating.

8.3.3 Tangential Developments

There are several ways to extend this work into fields not directly related to sentence recognition systems. Some of these are discussed in the following sections.

Identifying Keywords

We can use our structural tag classification system to identify some semantic keywords in a sentence which has already been recognised. By this method we can try to isolate and recognise subject domains, actors, actions, objects and events. We can do this by making defocusing frequency-dependent. High frequency words are highly defocused, whereas rare words are only slightly defocused. What results is a serial representation which contains single words — or small groups of semantically related words — between broad, low content classes.

Operating on Semantically Marked Data

The classification system described in this thesis can be used with objects other than uncomplicated words. We have already suggested that, following Brill *et al.*, the last three letters of English words might be good indicators of class information. The same could apply to semantic tags assigned to words : a mutual information algorithm could be applied to semantic tag collocations to find a coherent semantic classification system. If such a system existed, then mutual information could be used to find the best pairing of sentence and meaning structures, or sentence and action structures.

Morpheme Clustering

We have shown that the SAMI algorithm can discover patterns at the word level and at the phoneme level. We suggest that it should also be able to discover some morpheme-classes, given an appropriately constructed input corpus. This information could be useful in syntactic disambiguation; operating at the morphological level is also appealing because the set of morphemes of a language is smaller and grows at a slower rate than the set of words. Morphemes, in other words, might make better atomic units for statistical language processing than words. This is obvious if we consider an example. If the words ⟨`jumps`⟩, ⟨`jumping`⟩, ⟨`jumper`⟩ and ⟨`jumpers`⟩ only occur relatively infrequently in a corpus, a word-based statistical language processing system faces the sparse data problem. If, however, we consider the root ⟨`jump`⟩ separately from the various stems, then we alleviate the sparse data problem, since we have a single root which has a frequency equal to the sum of the original word frequencies, together with the common stems ⟨`s`⟩ ⟨`ing`⟩, *etc.*.

Information Retrieval Word Sense Disambiguation

If a user makes a database search for documents relating to the word ⟨`food`⟩, mutual information statistics could be used to prompt them about which sense or nuance of the word they meant — this might be achieved by accessing those words in the database for which the word and ⟨`food`⟩ have high mutual information : this set of words would allow the user to prune the search space.

Building a Single Class Database

Currently, there are as many class databases as there are classification levels. This can be improved upon greatly with some sophisticated database manipulation. In these databases, words are represented as TAGs. Words which are classified similarly should find themselves close to each other in the database, since, in the database, words are usually alphabetically stored. This has the following advantage. Imagine that we are searching for and find the 16-bit word bigram TAG $\langle 1110000000000000, 0111111111111111\rangle$; if we decide that this bigram does not occur with a frequency which satisfies our significance requirements, we might start looking for the frequency of the 15-bit bigram classification schema $\langle 111000000000000, 011111111111111\rangle$ of the tag. To find its frequency, we can just start our bidirectional search from the current point, for as long as each word bigram has the required 15-bit beginning. This method of searching would fit well with a back-off language model, and would be space efficient.

Mutual Information Parsing

Brill *et al.* have carried out some work on finding constituent boundaries by minimising a generalised mutual information equation. They use a pre-tagged corpus to generate statistics about word classes and this leads to a reasonably successful automatic parser. An interesting variation on this theme is to use a generalised mutual information minimiser on word classes which have been derived from the mutual information maximiser described in this thesis. Such a system would allow us to take any untagged natural language text and produce an entirely mutual information-based parse. This parse would almost certainly not be as linguistically instructive as a hand-parse, but it might show some other interesting surface features of the language.

Finitary Models as Components of Hybrid Cognitive Models

The finitary models upon which most of this research is based could be used as an informative adjunct to more cognitively plausible models — for example, they could help in resolving some ambiguities as illustrated by the three well-formed oronyms of figure 8.1. Generally, the use of segment probabilities can improve cognitive models since they allow certain syntactic constructions to be favoured over more arcane ones, they appear to offer sensible ways to model some selectional features of languages, and they suggest a mechanism whereby the vocabularies and word taxonomies of the various languages of the world, if not their syntax, could be learned.

8.4 Cognitive Relevance

The sentence

```
Oysters oysters oysters oysters oysters oysters oysters oysters
oysters oysters oysters oysters oysters oysters oysters oysters
split split split split split split split split split split split
split split split split split.
```

is, according to many linguists, well-formed. A finite-state grammar can be constructed to accept this sentence as well-formed, though, according to these linguists, such grammars cannot deal with sentences which are like this, but which have an indefinite degree of embedding. That is, for any finite-state grammar, a linguist could construct a whole family of sentences like the one above, only longer, so that the finite-state grammar fails to recognise

them as well-formed. This is considered a fatal theoretical limitation for finite-state grammars. Conversely, we have suggested in chapter one that, for any computer implementation of a grammar which is theoretically more powerful than finite-state ones, we can construct a finite-state grammar which is weakly equivalent. However, we admit that, from a cognitive modelling perspective, the more powerful grammar might be considered a better model of the human language processing ability. However, that humans have great difficulty deciding whether the example sentence above is well-formed or not, or, indeed just what the sentence means, we claim that language models which explicitly claim to be cognitive and yet easily accept these sentences as well-formed, are poor models of the human linguistic ability. Our point would have been better made with an 'oysters split' sentence (Pinker [124] calls them onion sentences) which was seventeen thousand words long, or perhaps one million and one words long, or perhaps with an acoustic version of that sentence. Similarly, cognitive models which never make performance errors or recognition errors — either acoustic or syntactic, for example oronyms and garden-path parses respectively — cannot be said to be coherent models of the human linguistic capacity either.

Grammars which seem cognitively plausible tend not to be as robust as stochastic finitary models; in this sense they fail as cognitive models. An improvement would be to incorporate the stochastic paradigm into grammar systems more powerful than finite-state ones. Implementing human-like constraints into a system which is based on, for example, a context-sensitive grammar, is one way of making these models more plausible.

That humans learn rather than inherit vocabulary items is undisputed. The same can be said of some classes of word. The mechanisms behind this acquisition process have not yet been uncontraversially stated.

Evidence that the human language processing ability makes use of n-gram statistics — not, of course as an addendum to a word-chain grammar model — might be offered in the form of certain types of 'garden path' sentence : ones which we cumulatively parse until we reach a point where our parse hypothesis breaks down and we are forced to back-track. Pinker notes that we could be drawn to the initially plausible parse by initially plausible (that is, relatively high probability) word pair constituents. Consider the following sentences, reproduced by Pinker :

```
The man who hunts ducks out on weekends.
The cotton clothing is usually made of grows in Mississippi.
The prime number few.
Fat people eat accumulates.
```

`The horse raced past the barn fell.`

The word pairs ⟨`cotton clothing`⟩, ⟨`hunts ducks`⟩, ⟨`prime number`⟩ and ⟨`fat people`⟩ might be common segments — their mutual information values could be so high that we automatically construct the inappropriate parse first.

Since one purpose of building computational models of natural language is to allow humans to communicate with computers in a more natural way, constructing a model of language as near to that of a human is an obvious necessity : we want the computer to speak in sentences which we can parse and understand easily; we want to speak hesitatingly and sometimes in ill-formed ways, and still hope that the computer system receives much of our original message; in short, we want as accurate a model of human language processing as possible, not just because we are interested in the mechanisms of human minds, but also because we want our engineering applications to be maximally familiar to use.

Appendix A

Syntax Based Classification Listing

AJ0 — adjective

AJC — comparative adjective

AJS — superlative adjective

AT0 — article

AV0 — unmarked adverb

AVC — comparative adverb

AVP — adverb particle

AVQ — wh-adverb

AVS — superlative adverb

CJC — coordinating conjunction

CJS — subordinating conjunction

CJT — the conjunction 'that'

CRD — cardinal numeral

DGE — possessive determiner

DT0 — general determiner

DTQ — wh-determiner

EX0 — existential 'there'

GEN — the genitive morpheme

ITJ — interjection or other isolate

NN0 — noun (neutral number)

NN1 — singular noun

NN2 — plural noun

NP1 — proper noun

NPP — special proper noun (placename)

ONE — the word ‘one’

ORD — ordinal

PGE — possessive (genitive) pronoun form

PNI — indefinite pronoun

PNP — personal pronoun

PNQ — wh-pronoun

PNX — reflexive pronoun

PRO — the preposition ‘of’

PRP — all other prepositions except ‘of’

PUN — punctuation

TO0 — infinitive marker

UNC — unclassified

VBB — base forms of the verb ‘be’

VBD — past form of ‘be’

VBN — ‘-ing’ form of ‘be’

VBZ — past participle of ‘be’

VDB — ‘-s’ form of ‘be’

VDD — base form of verb ‘do’

VDG — past form of ‘do’

VDN — ‘-ing’ form of verb ‘do’

VDZ — past participle of verb ‘do’

VHB — ‘-s’ form of verb ‘do’

VHD — base form of verb ‘have’

VHN — past form of verb ‘have’

VHG — ‘-ing’ form of verb ‘have’

VHZ — past participle of verb ‘have’

VM0 — ‘-s’ form of verb ‘have’

VVB — modal auxilliary verb

VVD — base form of lexical verb

VVG — ‘-ing’ form of lexical verb

VVN — past participle form of lexical verb

VVZ — ‘-s’ form of lexical verb

XX0 — the negative ‘not’ or ‘n’t’

ZZ0 — alphabetical symbol

Appendix B

Iterative Markov Model Transition Probability Re-Estimation

In section 7.2 we used an iterative update equation for training the parameters of an interpolated language model. In this appendix, we derive the equation. Consider figure B.1; if we assume for the moment that natural language text is generated by a Hidden Markov

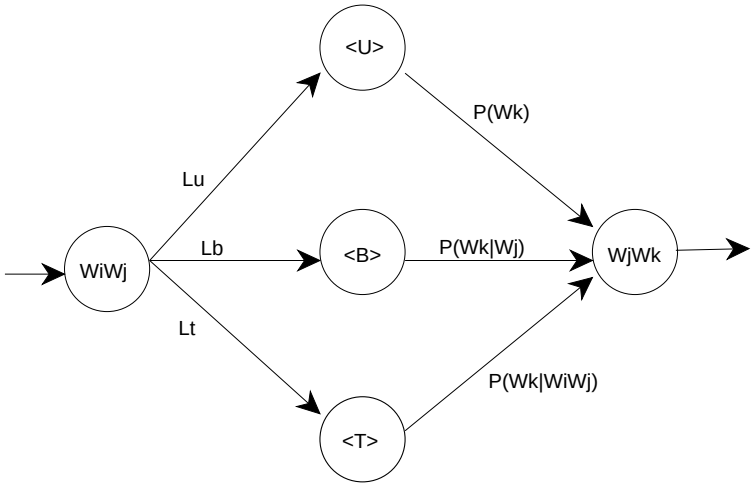


Figure B.1: Section of a Markov Chain showing the transition from the state corresponding to word-pair w_i,w_j to the state corresponding to word-pair w_j,w_k. The first three arcs, Lu, Lb and Lt correspond to the non-emitting unigram, bigram and trigram transition weights λ_u, λ_b and λ_t. The second set of arcs correspond to the maximum likelihood conditional probabilities of the word w_k, for unigram, bigram and trigram language models.

chain part of which is shown in this figure, and if we could observe the state of this chain, then we could estimate the non-emitting transition probabilities λ_u, λ_b and λ_t by counting how often state $\langle w_i, w_j \rangle$ is entered — $C(\langle w_i, w_j \rangle)$ — as well as how frequently the transition

$\langle w_i, w_j \rangle \rightarrow \langle U \rangle$ is made — $C(\langle w_i, w_j \rangle \rightarrow \langle U \rangle)$. Similar counts can be made for the transition $\langle w_i, w_j \rangle \rightarrow \langle B \rangle$ and $\langle w_i, w_j \rangle \rightarrow \langle T \rangle$, though we will now concentrate on estimating λ_u, as

$$\frac{C(\langle w_i, w_j \rangle \rightarrow \langle U \rangle)}{C(\langle w_i, w_j \rangle)}$$

Unfortunately, state transitions are hidden from us, so we cannot use the above expression. If, however, we already knew the weight λ_u, then we could find out the fraction of times, on average, that a word w is emitted after a $\langle w_i, w_j \rangle \rightarrow \langle U \rangle$ transition.

$$P(w \texttt{ following } \langle w_i, w_j \rangle \rightarrow \langle U \rangle) = \frac{\lambda_u \times P(w)}{P(w|\langle w_i, w_j \rangle)}$$

We can estimate $P(w|\langle w_i, w_j \rangle)$ immediately as

$$\lambda_u \times P(w) + \lambda_b \times P(w|w_i) + \lambda_t \times P(w|w_i, w_j)$$

so that

$$P(w \texttt{ following } \langle w_i, w_j \rangle \rightarrow \langle U \rangle) = \frac{\lambda_u \times P(w)}{\lambda_u \times P(w) + \lambda_b \times P(w|w_i) + \lambda_t \times P(w|w_i, w_j)}$$

This probability, which for the sake of simplicity we will call $p_u(w)$, is the fraction of times the model selects the unigram path through the part of the chain shown in figure B.1. Now all we need is some estimate of how often, on average, we reach state $\langle w_i, w_j \rangle$; this is provided by the observed count of the bigram pair $\langle w_i, w_j \rangle$ in the held-out corpus — $C_h(\langle w_i, w_j \rangle)$. If this held-out text is large enough, and we have some arbitrary initial estimate for λ_u, then we can improve upon this estimate by setting a new λ'_u based on estimated counts C_e as follows

$$\lambda'_u = \frac{C_e(\langle w_i, w_j \rangle \rightarrow \langle U \rangle)}{C_e(\langle w_i, w_j \rangle)} = C_h(\langle w_i, w_j \rangle) \times p_u(w)$$

In the general case, where there are L language models, the update equation for the jth transition weight is

$$\lambda'_j = \sum_{i=1}^{n} \frac{\lambda_j \times P_{LM_j}(w_i)}{\sum_{k=1}^{L} \lambda_k \times P_{LM_k}(w_i)}$$

and the sum is over all n word tokens of the held-out corpus.

Bibliography

[1] Philip E. Agre and David Chapman. What are plans for? *Robotics and autonomous systems*, 6:17–34, 1990.

[2] Hendrik James Antonisse. A grammar-based genetic algorithm. In *Foundations of Genetic Algorithms*, pages 193 – 204, 1991.

[3] Joseph J. Atick and A. Norman Redlich. Towards a theory of early visual processing. *Neural Computation*, 2:308 – 320, 1990.

[4] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183 – 193, 1954.

[5] Lalit R. Bahl, Peter F. Brown, Peter V. DeSouza, and Robert L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(7):1001 – 1008, July 1989.

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

[7] Alwyn Barry. The emergence of high level structure in classifier systems — a proposal. In R. Cowie and M. Owens, editors, *Proceedings of the Sixth Irish Conference on Artificial Intelligence and Cognitive Science*, pages 185 – 196, September 1993.

[8] Roberto Basili, Teresa Pazienza, and Paolo Velardi. Combining NLP and statistical techniques for lexical acquisition. In *Probabilistic Approaches to Natural Language*. American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[9] Roberto Basili, Teresa Pazienza, and Paolo Velardi. What can be learned from raw texts? *Machine Translation*, 8:147 – 173, 1993.

[10] R. Beale and T. Jackson. *Neural Computing : An Introduction.* Adam Hilger, 1990.

[11] R. Beckwith, C. Fellbaum, D. Gross, and G. Miller. WordNet: A lexical database organized on psycholinguistic principles. In Uri Zernik, editor, *Lexical Acquisition : Exploiting On-Line Resources to Build a Lexicon*, chapter 9, pages 211 – 232. Lawrence Erlbaum Associates, 1991.

[12] Robert C. Berwick. Learning from positive-only examples — the subset principle and three case studies. In J. C. Carbonell R. S. Michalski and T. M. Mitchell, editors, *Machine Learning : An Artificial Intelligence Approach (Volume 2).* Morgan Kaufmann Publishers, 1986.

[13] Rens Bod. A computational model of language performance. In *Fourth European Summer School in Logic Language and Information : Corpus Based Language processing*, 1992.

[14] Eric Brill, David Magerman, Mitchell Marcus, and Beatrice Santorini. Deducing linguistic structure from the statistics of large corpora. In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1990.

[15] Eric Brill and Mitch Marcus. Tagging an unfamiliar text with minimal human supervision. In *Probabilistic Approaches to Natural Language.* American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[16] Rodney A. Brooks. Intelligence without reason. report, 1991.

[17] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[18] Peter F. Brown, Vincent Della Pietra, Peter deSouza, Jennifer C. Lai, and Robert C. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467 – 479, 1992.

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

[20] R. Brown. *A First Language : The Early Stages.* Penguin, Harmondsworth, England, 1973.

[21] K. A. Brownlee. *Statistical Theory and Methodology in Science and Engineering.* John Wiley and Sons inc., 1965.

[22] John D. Burger and Dennis Connolly. Probabilistic resolution of anaphoric reference. In *Probabilistic Approaches to Natural Language.* American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[23] Glenn Carroll and Eugene Charniak. Learning probabilistic dependence grammars from labelled text. In *Probabilistic Approaches to Natural Language.* American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[24] G.J. Chaitin. Randomness and complexity in pure mathematics. *International Journal of Bifurcation and Chaos*, 4(1), February 1994.

[25] David J. Chalmers, Robert M. French, and Douglas R. Hofstadter. High level perception, representation and analogy : a critique of artificial intelligence methodology. Technical Report 49, Center for Research in Cognition and Computing, Indiana University, March 1991.

[26] Ye-So Chen. Statistical models of text in continuous speech recognition. *Kybernetes*, 20(5):29 – 40, 1991.

[27] Noam Chomsky. *Syntactic Structures.* The Hague: Mouton, 1957.

[28] Noam Chomsky. *Aspects of the Theory of Syntax.* M.I.T. Press, 1965.

[29] Philip A. Chou. Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):340 – 354, April 1991.

[30] K. Church, W. Gale, P. Hanks, and D. Hindle. Using statistics in lexical analysis. In Uri Zernik, editor, *Lexical Acquisition : Exploiting On-Line Resources to Build a Lexicon*, chapter 6, pages 115 – 164. Lawrence Erlbaum Associates, 1991.

[31] Kenneth W. Church and William A. Gale. A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilities of english bigrams. *Computer Speech and Language*, 5:19 – 54, 1991.

[32] Kenneth W. Church and Robert L. Mercer. Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics*, 19(1):1 – 23, 1993.

[33] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on applied Natural Language processing*, 1988.

[34] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information and lexicography. In *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, pages 76 – 82, 1989.

[35] Antoine Cohen. *The Phonemes of English — A Phonemic Study of the Vowels and Consonants of Standard English.* Martinus Nijhoff, 1965.

[36] S. Cookson. Final evaluation of VODIS. In *Proceedings of Speech '88, Seventh* FASE *Symposium*, pages 1311 – 1320, Edinburgh, 1988. Institute of Acoustics.

[37] Thomas M. Cover and Joy A. Thomas. *Elements of Information theory.* John Wiley and Sons, 1991.

[38] S. J. Cox. Hidden markov models for automatic speech recognition : Theory and application. In C. Wheddon and R. Linggard, editors, *Speech and Language Processing*, pages 209 – 230. Chapman and Hall, 1990.

[39] Jonathan Culler. *Saussure.* Fontana Press, 1976.

[40] Ferdinand de Saussure. *Course in General Linguistics.* Duckworth, 1983.

[41] Daniel Dennett. *Consciousness Explained.* London: Allen Lane, 1991.

[42] Anne-Marie Derouault and Bernard Merialdo. Natural language modelling for phoneme-to-text transcription. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), November 1986.

[43] Thomas G. Dietterich and Ryszard S. Michalski. Learning to predict sequences. In J. C. Carbonell R. S. Michalski and T. M. Mitchell, editors, *Machine Learning : An Artificial Intelligence Approach (Volume 2).* Morgan Kaufmann Publishers, 1986.

[44] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61 – 74, 1993.

[45] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14:179 – 211, 1990.

[46] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195 – 225, 1991.

[47] Jeffrey L. Elman. Incremental learning, or the importance of starting small. Technical Report 9101, Center for Research in Language, U.C.S.D., 1991.

[48] R. Fano. *Transmission of Information.* M.I.T. Press, 1961.

[49] R. D. Faulk. Segmenting discrete data representing continuous speech input. *I.B.M. Systems Journal*, 29(2), 1990.

[50] Steven Finch and Nick Chater. A hybrid approach to the automatic learning of linguistic categories. *A.I.S.B. Quarterly*, 1991.

[51] Steven Finch and Nick Chater. Bootstrapping syntactic categories using statistical methods. In *Background and Experiments in Machine Learning of Natural Language*, pages 229–235, 1992.

[52] Steven Finch and Nick Chater. Learning syntactic categories : A statistical approach. In M. Oaksford and G.D.A. Brown, editors, *Neurodynamics and Psychology*, chapter 12. Academic Press, 1994.

[53] Steven Paul Finch. *Finding Structure in Language.* PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1993.

[54] J. R. Firth. A synopsis of linguistic theory 1930 – 1955. In F. Palmer, editor, *Selected Papers of J. R. Firth.* Longman, 1968.

[55] David Fisher and Ellen Riloff. Applying statistical methods to small corpora : Benefitting from a limited domain. In *Probabilistic Approaches to Natural Language.* American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[56] Richard Forsyth and Roy Rada. *Machine Learning : Applications in Expert Systems and Information Retrieval.* Ellis Horwood, 1986.

[57] W. Nelson Francis and Henry Kucera. *Frequency Analysis of English Usage: Lexicon and Grammar.* Houghton Mifflin, Co., Boston Mass., 1982.

[58] William A. Gale and Kenneth W. Church. Poor estimates of context are worse than none. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 283 – 287, 1990.

[59] William A. Gale, Kenneth W. Church, and David Yarwosky. Work on statistical methods for word sense disambiguation. In *Probabilistic Approaches to Natural Language.*

American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[60] Alan Garnham. *Psycholinguistics — Central Topics.* Methuen, 1985.

[61] Roger Garside, Geoffrey Leech, and Geoffrey Sampson, editors. *The Computational Analysis of English, a Corpus Based Approach.* Longman, London, 1987.

[62] Michael Gasser. Learning syllable representations : A connectionist approach. In *The Cognitive Science of Natural Language Processing*, 1992.

[63] Steven Gillis. Topics in 'natural' natural language acquisition. In *Background and Experiments nin Machine Learning of Natural Language*, pages 25—52, 1992.

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

[65] D.E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison Wesley, 1989.

[66] Rafael C. Gonzalez and Michael G. Thomason. *Syntactic Pattern Recognition.* Addison Wesley, 1978.

[67] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, December 1953.

[68] A.L. Gorin, S.E. Levinson, A.N. Gertner, and E. Goldman. Adaptive acquisition of language. *Computer Speech and Language*, 5:101 – 132, 1991.

[69] Joseph H. Greenberg. *Essays in Linguistics.* University of Chicago Press, 1957.

[70] Gregory Grefenstette. Finding semantic similarity in raw text : The deese antonyms. In *Probabilistic Approaches to Natural Language.* American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[71] Ralph Grishman and John Sterling. Acquisition of selectional patterns. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING'92)*, pages 658 – 664, 1992.

[72] J. Gumprez. The speech community. In Pier Paolo Giglioli, editor, *Language and Social Context*, pages 219–231. Pelican, 1972.

[73] V. Gupta, M. Lennig, and P. Mermelstein. A language model for very-large vocabulary speech recognition. *Computer Speech and Language*, 6:331 – 344, 1992.

[74] Young S. Han, Young C. Park, and Key-Sun Choi. Recursive markov chain as a stochastic grammar. In *Probabilistic Approaches to Natural Language*. American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[75] Zellig S. Harris. *Structural Linguistics*. Phoenix Books, 1951.

[76] James Higginbotham. Noam Chomsky's linguistic theory. In Steve Torance, editor, *The Mind and the Machine : Philosophical aspects of Artificial Intelligence*, chapter 8. Ellis Horwood, 1984.

[77] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[78] J. H. Holland. Escaping brittleness : The possibilities of general purpose learning algorithms applied to parallel rule-bases systems. In J. G. Carbonell R. S. Michalski and T. M. Mitchell, editors, *Machine Learning II*. Morgan Kaufmann, 1986.

[79] John Hughes. *Automatically Acquiring a Classification of Words*. PhD thesis, School of Computer Studies, University of Leeds, 1994.

[80] John Hughes and Eric Atwell. The automated evaluation of inferred word classifications. In *Eleventh European Conference on Artificial Intelligence*, 1994.

[81] Jonathan J. Hull. Combining syntactic knowledge and visual text recognition : A hidden markov model for part of speech tagging in a word recognition algorithm. In *Probabilistic Approaches to Natural Language*. American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[82] D. Hymes. Towards ethnographies of communication : The analysis of communicative events. In Pier Paolo Giglioli, editor, *Language and Social Context*, pages 21–44. Pelican, 1972.

[83] Frederick Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the I.E.E.E.*, 64(4), April 1976.

[84] Frederick Jelinek. The development of an experimental discrete dictation recogniser. *Proceedings of the I.E.E.E.*, 73(11), 1985.

[85] Frederick Jelinek. Self-organised language modeling for speech recognition. Technical report, I.B.M. Continuous Speech Recognition Group, 1985.

[86] Frederick Jelinek, Robert L. Mercer, and Salim Roukos. Principles of lexical language modelling for speech recognition. In S. Furui and M.M. Sondhi, editors, *Advances in Speech Signal Processing*. Maral Dekku, Inc., 1992.

[87] S.J. Johansson, E.S. Atwell, R. Garside, and G. Leech. *The tagged LOB Corpus : Users' Manual*. The Norwegian Centre for the Humanities, Bergen, 1986.

[88] M. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 531 – 546. Lawrence Erlbaum Associates, 1986.

[89] Slava M. Katz. Estimation of probabilities for sparse data for the language model component of a speech recogniser. *I.E.E.E. Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400 – 401, March 1987.

[90] Roger M. Keesing. *Cultural Anthropology, a Contemporary Perspective*. Holt, Rinehart and Winston, New York, 1976.

[91] Clyde Kluckhohn. *Mirror for Man*. McGraw-Hill Book Company, 1949.

[92] Reinhard Kneser and Hermann Ney. Forming word classes by statistical clustering for statistical language modelling. In R. Köhler and B.B. Rieger, editors, *Contributions to Quantitative Linguistics*, pages 221 – 226. Kluwer Academic Publishers, 1993.

[93] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1:4 – 7, 1964.

[94] J. R. Koza. Hierarchical genetic algorithms that operate on populations of computer programs. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 768 – 780, 1989.

[95] Ronald Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570 – 583, June 1990.

[96] Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language*, 6:225 – 242, 1992.

[97] K. Lari and S.J. Young. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 5:237 – 257, 1991.

[98] Stephen E. Levinson. Structural methods in automatic speech recognition. In *Proceedings of the I.E.E.E. Volume 73, Number 11*, 1985.

[99] Roger Lewin. *Complexity.* Phoenix Books, 1993.

[100] Mark K. Liberman. The trend towards statistical models in natural language processing. In *Natural Language and Speech Symposium Proceedings*, 1991.

[101] Elizabeth D. Liddy and Woojin Paik. Statistically-guided word sense disambiguation. In *Probabilistic Approaches to Natural Language.* American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[102] Ralph Linsker. Self-organization in a perceptual network. *I.E.E.E. Computer*, 21(3):105 – 117, 1988.

[103] John Lyons. *Chomsky.* Fontana, 1977.

[104] Bruce MacLennan. Synthetic ethology : An approach to the study of communication. In Christpoher G. Langton, Charles Taylor, J. Doyne Farmer, and Steen Rasmussen, editors, *Artificial Life II : A Proceedings Volume in the Santa Fe Institute for Studies in the Sciences of Complexity*, pages 631–658. Addison-Wesley, 1991.

[105] John Makhoul, Fred Jelinek, Larry Rabiner, Clifford Weinstein, and Victor Zue. Spoken language systems. *Annual Review of Computer Science*, 4:481 – 501, 1990.

[106] Bronislaw Malinowski. The problem of meaning in primitive languages. In C. K. Ogden and I. A. Richards, editors, *The Meaning of Meaning*, page 307. Routledge and Kegan Paul, 1949.

[107] Bernard Manderick. The genetic algorithm. In *Background and Experiments in Machine Learning of Natural Language*, 1992.

[108] M. McGee-Wood. *Categorial Grammars.* Routledge, London, 1993.

[109] John McMahon and F. J. Smith. Structural tags, annealing and automatic word classification. Unpublished Report, Queen's University of Belfast.

[110] Ray Meddis. *Statistics Using Ranks — A Unified Approach.* Basil Blackwell, 1984.

[111] George A. Miller. *Language and Communication.* New York: McGraw-Hill, 1951.

[112] George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1 – 28, 1991.

[113] Robert H. Moore. Contexts. In Wallace L. Anderson and Norman C. Stageberg, editors, *Introductory Readings on Language*. Holt, Rinehart and Winston Inc., 1970.

[114] Authur Nádas. Estimation of probabilities in the language model of the IBM speech recognition system. *I.E.E.E. Transactions on Acoustics, Speech and Signal Processing*, ASSP-32(4), August 1984.

[115] Sven Naumann and Jürgen Schrepp. Inductive learning of reversible grammars. In *Background and Experiments in Machine Learning of Natural Language*, pages 237–243, 1992.

[116] Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1 – 38, 1994.

[117] John S. Nicholis. *Chaos and Information Processing — A Heuristic Outline*. World Scientific, 1991.

[118] John S. Nicholis. Chaotic dynamics of linguistic-like processes at the syntactical and semantic levels : In pursuit of a multifractal attractor. In *Patterns, Information and Chaos in Neuronal Systems*. World Scientific, 1993.

[119] Peter O'Boyle. *A Study of an N-gram Language Model for Speech Recognition*. PhD thesis, Department of Computer Science, Queen's University, Belfast, 1993.

[120] Peter O'Boyle. A statistical language model for speech recognition. In *IASTED International Conference Artificial Intelligence, Expert Systems and Neural Natworks*, Zurich, Switzerland, July 1994.

[121] Peter O'Boyle, Marie Owens, and F.J. Smith. A weighted average model of natural language. Computer Speech and Language — Forthcoming.

[122] Peter O'Boyle and F.J. Smith. Markov models of natural language. Unpublished Report.

[123] Fernando Pereira and Naftali Tishby. Distributed similarity, phase transitions and hierarchical clustering. In *Probabilistic Approaches to Natural Language*. American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[124] Steven Pinker. *The Language Instinct — The New Science of Language and Mind.* Allen Lane, Penguin Press, 1994.

[125] M. D. Plumbley. Information theory and neural network learning algorithms. In Gerry Orchard, editor, *Neural Computing – Research and Applications*, pages 145 – 155. Institute of Physics Publishing, 1993. Proceedings of the Second Irish Neural Networks Conference.

[126] J. B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46:77 – 105, 1990.

[127] K. R. Popper. *The Logic of Scientific Discovery.* London: Hutchinson, 1959.

[128] David Powers and Walter Daelemans. SHOE : The extraction of hierarchical structure for machine learning of natural language (project summary). In *Background and Experiments in Machine Learning of Natural Language*, pages 125–161, 1992.

[129] L. R. Rabiner and B. J. Juang. An introduction to hidden markov models. *I.E.E.E. A.S.S.P. Magazine*, pages 4 – 16, January 1986.

[130] Allan Ramsay. Linguistics : The cognitive science of natural language. In *Third Conference on the Cognitive Science of Natural Language Processing.* Dublin City University, July 1994.

[131] Martin Redington, Nick Chater, and Steven Finch. Distributional information and the acquisition of linguistic categories : A statistical approach. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 1993.

[132] Martin Redington, Nick Chater, and Steven Finch. The potential contribution of distributional information to early syntactic category acquisition. Unpublished Report, 1994.

[133] Ronan Reilly. A connectionist technique for on-line parsing. In *The Cognitive Science of Natural Language Processing*, 1992.

[134] Ronan Reilly. An exploration of clause boundary effects in simple recurrent network representations. In *The Second Irish Neural Netowrks Conference*, 1992.

[135] Philip S. Resnik. *Selection and Information : A Class-Based Approach to Lexical Relationships.* PhD thesis, Computer and Information Science, University of Pennsylvania, December 1993. Institute for Research in Cognitive Science Report I.R.C.S.-93-42.

[136] Jan Robin Rohlicek, Yen-Lu Chow, and Salim Roucos. Sytatistical language modelling using a small corpus from an application domain. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 267 – 270, 1988.

[137] R. Rucker. *Mind Tools — The Mathematics of Information.* Penguin, 1988.

[138] Arto Salomaa. Probabilistic weighted grammars. *Information and Control*, 15:529 – 544, 1969.

[139] Geoffrey Sampson. Evidence against the Grammatical/Ungrammatical distinction. In Wilem Meijs, editor, *Corpus Linguistics and Beyond — Proceedings of the Seventh International Conference on English Language Research on Computerized Corpora*, pages 219 – 226. Rodopi,Amsterdam, 1987.

[140] Geoffrey Sampson. The need for grammatical stocktaking. In Rens Bod and Remko Scha, editors, *Corpus Based Language Processing Workshop.* Fourth European Summer School in Logic, Language and Information, 1992.

[141] Edward Sapir. Language defined. In Wallace L. Anderson and Norman C. Stageberg, editors, *Introductory Readings on Language*, pages 1–12. Holt, Rinehart and Winston, 1970.

[142] J. C. Scholtes. Resolving linguistic ambiguities with a neural data oriented parsing (dop) system. In *Background and Experiments in Machine Learning of Natural Language*, pages 279–282, 1992.

[143] Hinrich Schütze. Part-of-speech induction from scratch. In *Proceedings of the Association for Computational Linguistics 31*, pages 251 – 258, 1993.

[144] C.E. Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 1951.

[145] R. A. Sharman. Evaluating a grammar as a language model for speech. In E. Masgrau L. Torres and M. A. Lagunas, editors, *Signal Processing V : Theories and Applications.* Elsevier Science Publishers, B. V., 1990.

[146] George W. Smith. *Computers and Human Language.* Oxford University Press, 1991.

[147] D. Solomon. Learning a grammar. Technical Report UMCS-AI-91-12-1, University of Manchester Department of Computer Science, 1991.

[148] D. Solomon and M. McGee-Wood. Unified lexicon and grammar. In Russell J. Collingham, editor, *Workshop on the Unified Lexicon*, December 1993.

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

[150] Rohini Srihari and Charlotte M. Balthus. Combining statistical and syntactic methods in recognising handwritten sentences. In *Probabilistic Approaches to Natural Language*. American Association for Artificial Intelligence, AAAI Press, 1992. Technical report FS-92-05.

[151] Jane Stephens and Geoffrey Beattie. Turn-taking on the telephone : Textual features which distinguish turn-final and turn-medial utterances. *Language and Social Psychology*, 5(3):211 – 222, 1986.

[152] George Sugihara and Robert M. May. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344:734 – 741, 1990.

[153] Richard F.E. Sutcliffe, Annette McElligott, and G. O'Néill. Irish-English lexical translation using distributed semantic representations. In R. Cowie and M. Owens, editors, *Artificial Intelligence and Cognitive Science*, 1993.

[154] Michael K. Tanenhaus. Psycholinguistics : An overview. In Frederick J. Newmeyer, editor, *Linguistics : The Cambridge Survey*, volume III, chapter 1, pages 1–37. Cambridge University Press, 1988.

[155] M. Mitchell Waldrop. *Complexity : The Emerging Science at the Edge of Order and Chaos*. Viking, 1993.

[156] Mark Allen Weiss. *Data Structures and Algorithm Analysis in C*. The Benjamin/Cummings Publishing Company, 1993.

[157] Sandra Williams. (Semi-)Automatically acquiring lexical entries. Master's thesis, Cambridge University, August 1991.

[158] Ludwig Wittgenstein. *Philosophical Investigations*. Blackwell, 1953.

[159] J. Gerard Wolff. *Towards a Theory of Cognition and Computing*. Ellis Horwood, 1991.

[160] J. Gerard Wolff. Language learning, cognition and computing : A summary. In *Background and Experiments in Machine Learning of Natural Language*, 1992.

[161] P. J. Wyard and C. Nightingale. Grammar recognition by a single layer higher order neural net. *B.T. Technological Journal*, 10(3), 1992.

[162] David Yarowsky. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 454 – 460, 1992.

[163] Uri Zernik. Introduction. In Uri Zernik, editor, *Lexical Acquisition : Exploiting On-Line Resources to Build a Lexicon*, chapter 1, pages 1 – 26. Lawrence Erlbaum Associates, 1991.

[164] G. K. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.