Predicting query types by prosodic analysis

by

Simon Graham Jeremy Smith

A thesis submitted to

The University of Birmingham

for the degree of

DOCTOR OF PHILOSOPHY

School of Electronic, Electrical and Computer Engineering

The University of Birmingham

August 2003

Abstract

A body of work exists on the classification, by prosodic analysis and other means, of utterance types and dialogue moves in spoken corpora. Much of this output, while often linguistically well motivated, tends to rely on hand-crafted rules. This thesis presents a data-driven approach to the classification of utterances, using a novel combination of existing algorithmic approaches. Previous work has generally classified utterances according to such categories as wh- question, yes/no question, acknowledgement, response and the like; in general, the audio data used has been specially commissioned and recorded for research purposes.

The work presented here departs from this tradition, in that the recorded data consists of genuine interaction between the telephone operator and members of the public. Moreover, most of the calls recorded can be characterized as queries. The techniques presented in this thesis attempt to determine, automatically, the class of query, from a set of six possibilities including "statement of a problem" and "request for action". To achieve this, a scheme for automatically labelling utterance segments according to their prosodic features was devised, and this is presented. It is then shown how labelling patterns encountered in training data can be exploited to classify unseen utterances.

Acknowledgements

Thanks go first and foremost to Martin Russell for his patience, support and valuable insight as my supervisor over my time at Birmingham.

I'd like to thank my present and former colleagues in the speech group for being so encouraging and helpful. Specific technical assistance came from Philip Jackson, Nick Wilkinson and Paul Dixon, while Jin Jian-hong, Lo Boon-hooi and David Moreno provided me with software implementations which were needed for my experiments.

A big thank you goes to all those who participated in the psycholinguistic experiment described in Chapter 6.

The help of the following people for providing advice and resources is also gratefully acknowledged: David Attwater and colleagues, at BTexaCT, for providing the Oasis corpus, as well as much useful advice; Jane Morgan, of the School of Psychology in this university, for assistance in the design of the psycholinguistic experiments of Chapter 6; Daniel Hirst of the University of Provence, for providing the MOMEL and INTSINT implementations, and explaining how they worked; Helen Wright-Hastie and Massimo Poesio of Edinburgh University, for their support and advice; Mark Huckvale of University College London for answering questions on SFS (nearly all of which is Mark's work); and Harvey Lloyd-Thomas of Ensigma Technologies, for assistance with the BBC news corpus discussed in Chapter 2.

1. Introduction

This thesis explores the relationships between the message a speaker conveys in an utterance, and features of the utterance at a particular linguistic stratum: that of prosody. It is quite easy to demonstrate a correspondence at a trivial level. One of the acoustic correlates of prosody is **intensity**, or energy, and it is clearly possible to register anger, for example, or emphasis, by speaking more loudly (or shouting) during all or part of an utterance. Another acoustic correlate is **duration**: here, unusually drawn-out pronunciation might reflect frustration on the part of the speaker, while in the domain of **pitch modulation**, a question to which the possible answers are *yes* or *no* is often characterized by a rise in pitch towards the end of the utterance.

The work described in this thesis is chiefly concerned with the pitch characteristics of speech. The thesis describes a range of experiments which attempt to classify unseen spoken queries according to their type (**utterance type**, abbreviated to UT throughout the thesis). The initial phase of the work, the **fundamental frequency** (F_0) analysis, yields a parsimonious set of labels which capture prosodic features of the utterances. A subsequent phase of analysis shows that the assignment of particular labels, or sequences of labels, may characterize an utterance as more likely to belong to a particular UT class than any other.

The main purpose of this thesis, then, is to give an account of the design and architecture of a highly parametrizable suite of programs which accepts, as input, speech data which has been pre-annotated by UT, and outputs a predicted UT for utterances in unannotated test data. The suite of programs is known as **PLoNQ** (Prosodic Labelling of Natural Queries).

1.1 Motivation for the approach

In linguistics, **speech act theory** (Searle & Vanderveken 1985, Austin 1962) proposes categories to which utterances may be assigned in terms of the intent of the speaker, such as **declarative**, **interrogative** and so forth. It is accepted by researchers in the field – and indeed by the wider public – that there are strong affinities between speech act type (or UT) and prosodic effects, including pitch modulation.

Other researchers have tackled the problem of utterance classification by prosodic and indeed other linguistic means: examples are discussed in Chapter 2. A feature that earlier approaches share is that they rely on the handcrafting of rules, and on manual annotation of speech corpora with prosodic labels, syllable boundaries and the like. The system described here, by contrast, is data-driven: the annotation and subsequent utterance classification are entirely automatic, the only manual intervention required being the supply of values for various parameters in system tuning.

In most other reported work, the speech data is not entirely natural. It is spontaneous in the sense that it is not simply being read out, but in most cases the subjects being recorded were aware that the recordings were taking place, and the reason they were speaking at all was precisely to compile the corpus concerned. We were fortunate in being given access to a corpus of genuine, spontaneous speech. Notable also is that the type of speech available to us for analysis consisted of queries, and that our task was to classify them by type of query – "request for information", "statement of problem" and so on. This sort of classification task is more closely allied to what might be required in a real application, such as query facilities on a personal digital assistant, for example, or call routing, than casting a particular utterance as an *acknowledgement*, say, or a *backchannel*, classes available in the Edinburgh Map Task corpus (Anderson et al 1991).

The algorithms employed in the work are not new. The specific combination of algorithmic approaches is, however, novel; and for the most part it was implemented by the author, in a combination of C programs and UNIX scripts, with much use also of SFS (Speech Filing System) and its SML (Speech Measurement Language) (Huckvale & University College London 2001). Some additional material was written in Matlab, DOS and TCL/TK. Where another person is responsible for an implementation, this is made clear in the course of the thesis.

1.2 Organization of the thesis

After the present introductory chapter, a number of classification systems devised by other researchers are surveyed. Some, such as Wright (2000), tackle a fairly similar problem to that presented here, extracting prosodic and other linguistic data from utterances in order to assign them to one category; in other words, to determine UT. Related work, such as that of Nagata & Morimoto (1993), is also reported, even though their classification depends not on prosodic

features, but discourse context information.

The chapter treats utterance classification as a special case of a more general multi-class problem, and investigates earlier work done on **language identification** by prosodic and other means. Topic identification experiments conducted by the author are also reported. These experiments use a corpus of news stories, and exploited the vocabulary as the sole source of linguistic information, attempting to determine the news category (foreign, sports and so on) to which each unseen story belonged.

The third chapter describes the data used in the principal set of experiments that the thesis reports on. The format and structure of the BTexaCT Oasis corpus (BTexaCT 2001) are discussed here: the utterance types available and the rules for assigning them, as well as details of the annotation of the corpus. One or two limitations of the corpus are pointed out too.

Turning to the system architecture: the software consists of a number of algorithm implementations, discrete modules that are piped together to form PLoNQ, the UT classifier system. Principal components are: the **extraction** of a fundamental frequency contour from the raw sound file; the **segmentation** and **stylization** of that contour to eliminate micro-prosodic effects; the **discretization** stage, where segments are assigned to one of a closed set of prosodic labels; the **n-gram module**, which tries to ascertain which sequences of prosodic labels, when encountered in an observed utterance, are likely to characterize a particular UT. Finally, there is the **classifier module** itself. The fourth chapter describes all these modules, comparing the implementations to the relevant work of others. Details of certain user-specifiable system parameters (such as constraints on segment length, and the number of available prosodic labels) are also presented in this chapter.

Chapter 5 is the core chapter of the work, as it describes the experiments that were conducted and sets out the results. In this chapter, attention is paid to results at all experimental stages; a critical path is navigated through the several modules, and optimal parameter values are established, as the discussion proceeds. It was deemed more important to justify the use of particular parameter settings on intuitive or empirical grounds, than to conduct an exhaustive set of experiments whereby the best settings could only be determined when the final classification results for each

case were known. Thus, in deciding what sort of constraints should be applied to segment length, the motivation sprang as much from visual inspection and reflection on what constituted a linguistically meaningful segment as from overall performance.

Chapter 6 describes a psycholinguistic experiment: it investigates the performance of human subjects in what is broadly the same classification task as the principal research endeavour. In general, in this work, a suite of computer programs is expected to classify utterances on the basis of prosodic information alone. The training stimulus, it could be argued, is fairly impoverished; perhaps so impoverished that our performance expectations are unreasonable. It might as well be said straight away – and it will probably come as no surprise to the reader – that no near-perfect results are going to emerge from this thesis, and we will be looking essentially to find the best possible improvement over chance performance. If the task is difficult for a computer, it is interesting to know how challenging people would find it. The experiment was conducted, in part, by playing to subjects sound recordings which had been disguised so that only prosodic information was, in principle, available to them.

There are a number of different schemes for the prosodic annotation of speech. Many are more suitable for manual transcription than the automatic treatment of large amounts of data, and there is some discussion of this in the remaining sections of the present chapter. The pitch contour stylization and discretization systems **MOMEL** and **INTSINT**, designed by Daniel Hirst of the Université de Provence, are suitable for automatic annotation, though, and it was decided to use this software to label the Oasis data. The results were then passed to the PLoNQ n-gram and classifier modules, in order to compare the relative performance of the two labelling approaches. This comparative work is the subject of Chapter 7.

The final chapter returns to the central question posed by the thesis: how effectively may UT be assigned to unseen utterances on prosodic evidence? A single experiment is performed, using held-out data, and the parameter settings that were established as optimal in Chapter 5. The reason for the use of a new set of test data is to safeguard the system against a potential objection that the parameter settings were in some way tuned to the original test data. The reader will also find general conclusions in this chapter, including a discussion of the limitations of this work, and of

the possibilities for further research.

1.3 Prosody: some definitions

Couper-Kuhlen (1986:1) tells us that the equivalent of the term *prosody* among the ancient Greeks referred initially to the "tone or melodic accent" associated with a word, then later to tone diacritics placed over syllables. Later still, the tonal accents of Ancient Greek disappeared, and by the Middle Ages, the term had become associated primarily with the versification of poetry, where it might be more specifically named *metrical prosody*. It has retained that primary meaning to this day; but, for the speech scientist and linguist at least, it seems to have regained something of its original classical meaning. A useful working definition might be: the non-segmental components of the speech stream, including – at the perceptual level – pitch modulation, loudness, speaking rate and rhythm.

The same author discusses **paralinguistic** and **non-linguistic** non-segmental phenomena, and asserts that they are *not* in the realm of prosody. She identifies the former type as a temporary modification of the voice to accommodate certain circumstances, such as laughing, sobbing or speaking in a whisper. Certainly these have a communicative function, but their distinguishing feature, according to Couper-Kuhlen, is that they arise in speech only sporadically, whereas prosodic effects are always present. Others might prefer to argue that laughing and sobbing are ruled non-linguistic because they don't involve speaking at all; and that whispering, a process which is activated by speaking without vibration of the vocal folds, even for what would otherwise be voiced segments, is very much a part of prosody, as is speaking in an unusually low voice. Examples of non-linguistic events are those which the speaker does not have control over: sneezing, coughing and, following Roach (1991:133), the rapid rises and falls in pitch that obtain when one is simultaneously speaking and riding fast on a horse.

Kerbrat-Orecchioni (1977:58) defines prosody, slightly more restrictively than the working definition given above, as "l'ensemble des traits suprasegmentaux qui, durant l'émission vocale, se surajoutent à la chaîne phonique sans en épouser le découpage en phonèmes".[1]

Kerbrat-Orecchioni's definition is slightly at odds with that of Crystal (1969:5), who applies a further restriction. He explains prosody as "sets of mutually defining phonological features which have an essentially variable relationship to the words selected": for Kerbrat-Orecchioni, but not for Crystal, these features may serve to disambiguate semantically divergent utterances which share a segmental realization, such as the French examples in (1.1), (1.2) and (1.3).

(1.1) Quelle tombe! (What a tomb!)

(1.2) Quelle tombe? (Which tomb?)

(1.3) Qu'elle tombe! (May she fall!)

Kerbrat-Orecchioni's sentences are rather contrived, but they do illustrate the point. (1.3) has no lexical items in common with the other two utterances. (1.1) and (1.2) do share the noun *tombe*, and while the two instances of *quelle* may realize the same lexical entry, this is a case of polysemy, as one is an interrogative, the other an intensifier.

Similarly, prosodic phenomena may help to distinguish between homophones, in very much the same way as minimal pairs in segmental phonology. For example, while the two phones [r] and [l] are used to distinguish such pairs of words as *rice* and *lice*, *fry* and *fly* in English, and thus are accorded phonemic status, in Japanese they are two potential realizations of the same phoneme. Similarly discriminable pairs are found in lexical stress patterning, as manifested by variation in pitch and loudness. Slightly tenuous examples often given are verb/noun homonyms such as *contrast*, *record* and *segment*, where the stress shifts according to the part of speech in context – tenuous, because the unstressed syllable also undergoes vowel reduction to either [I] or schwa

[1] The set of suprasegmental features which, during speech production, may be said to overlay the phonemic sequence without being constrained to the same segmentation.

where this is possible, so that the grammatical conversion is also realized phonetically. The verb/noun *transform* is one case where the alternation is entirely prosodic in nature, however, while *plátano* and *platáno* (meaning “banana” and “plane-tree”, respectively, in Spanish) constitute another minimal pair, cited by Farinas (1999). Here, the distinction is made by the placement of the tonic accent.

1.3.1 Tone languages

With most Indo-European languages, then, we have to search quite hard for cases where prosodic detail is a prerequisite for clarity. This does not apply to the tone languages (a group which, according to Laver (1994:465) and Fromkin (1978:1) comprises most of the world's languages). In tone languages, prosodic features standardly play a semantic role, as well as conveying intonation patterns. In Lingala, a language of the Congo, there are minimal pairs which illustrate the grammatical function of tone, as shown in (1.4), (1.5) and (1.6).

(1.4) nàlámbá *I have prepared*

(1.5) nálámbà *I prepare*

(1.6) nàlámbà *I will prepare*

(Bwantsa-Kafungu 1972:17)

As with the majority of African tone languages, tonal distinctions are realized principally as differences in pitch **level** (Abercrombie 1967:109), and are identified as, for example, *high*, *low* and *mid*. Two things need to be made clear with respect to tone languages: first, the level of a given tone is relative to that of the other tones in the inventory, not to some absolute value – just as in tone and non-tone languages alike, pitch variation is determined by the dynamic range or **tessitura** of the individual who is speaking. Secondly, the relative tone levels must be interpreted as pitch targets, rather than fixed points: switching from tone to tone is no more abrupt than change of tongue posture in articulatory phonetics.

Mandarin (along with other varieties of Chinese), is generally designated a **contour** tone language, because the several tones are described in terms of pitch movement. The four principal tones of Mandarin are *level*, *rise*, *fall-rise* and *fall*; there is also a *neutral* tone, but this is only applied to grammatical particles and certain unstressed syllables. The tonal distinctions affect meaning at the lexical level, as may be seen from the following examples.

(1.7) ma➲ *mother* (1st tone)

(1.8) ma➲ *cannabis* (2nd tone)

(1.9) ma→ *horse* (3rd tone)

(1.10) ma➲ *scold* (4th tone)

(1.11) ma question particle (neutral)

In Mandarin, the smallest unit of meaning, or **morpheme,** is realized as a syllable (as a single character, incidentally, in the writing system). The inventory of syllables is impoverished, with about 400 syllable-types (as opposed to the thousands available in English). The function of the tone system, in terms of language evolution, is clear: a potential four-fold increase in the number of syllable-types. In practice, by no means all of the possibilities are taken up, and even taking account of tonal distinctions, there is still massive homophony in the language. One online dictionary, for example, lists 2 meanings for *ma* on the first tone, three on the second, five on the third, one on the fourth, and three on neutral.

1.3.2 Intonation and prosody

As a technical term, *intonation* seems only to complicate matters. Roach (1991) treats it as synonymous with *prosody*, while Crystal (1969:195) extends it to certain of the paralinguistic phenomena, such as whispering, at the same time noting that most authors use it to refer only to matters of pitch. Johns-Lewis (1986b) concurs, noting however that certain prosodic phenomena related to silence, such as "the relative amount of time occupied by articulatory activity as opposed to non-activity", do not belong to the realm of intonation. She indicates, too, that *prosody* is more applicable to models incorporating perceptual analysis, while the term *intonation* lends itself better to more quantitative experiments.

Let us briefly return to the tone languages. Some writers use the term **intonation language** to describe non-tone languages such as English (Abercrombie 1967; Roach 1991). This usage appears to be faulty, as it implies that tone languages are somehow bereft of intonation; as was hinted in the earlier discussion, tone applies at the syllable level, intonation to other levels up to and including the utterance. If the experiments that are to be reported in this thesis had been conducted on a corpus of Chinese, one would have been obliged to use the term *prosody*, as it would have been impossible to uncouple tone from intonation in the acoustic analysis. Since this research is in principle extensible to all languages, *prosody* is the term that will be used.

The PLoNQ data is from English, not Chinese. But prosody is a complex system, even in a non-tone language. Prosodic variations operate at all sub-utterance levels: they register lexical stress as well as emphatic stress, the overall intonation contour of the utterance as well as the phenomenon of **declination** or **downdrift** (the tendency, described by Pike (1948), for pitch to attenuate over the course of an utterance). As King (1998:98) points out, prosody encompasses everything "from segment type to the speaker's mood."

King's thesis was on speech recognition, and investigated the integration of a prosodic component into an ASR system. Precisely because of the complexity of prosody, and the multiplicity of tiers at which it operates, he eschewed prosodic analysis at the segmental level. Instead, he used the

utterance-wide pitch contour to identify UT, which enabled him to prefer one ASR result over another in cases of ambiguity (as for example in (1.1), (1.2) and (1.3) above).

1.3.3 The parameters of prosody

In the working definition above, loudness, speaking rate, rhythm and pitch movement were given as suitable perceptual prosodic parameters. In a data-driven, experimental project, it is necessary to take measurements of the acoustic correlates of these features.

As to loudness, most experimenters find root mean square (RMS) energy to be a more effective representation of the perceptual parameter than raw signal amplitude. It is computed "by taking the square root of the total energy in the amplitude spectrum from the short time Fourier analysis of the speech" (Wang & Seneff 2001) and then normalized to offset the effect of entire utterances being at different volumes.

Wang & Seneff's aim was to improve speech recognition performance by modelling lexical stress; theirs is a microscopic treatment, directly comparing stressed and unstressed syllables. For them, loudness, or its acoustic correlate, is highly relevant. In an analysis of stress in Welsh, Williams (1986) found that energy was a more significant index of Welsh lexical stress than even **fundamental frequency** (F_0), the acoustic correlate of pitch, which is of course central to prosodic analysis.

Utterance-wide approaches tend not to benefit from an energy contribution. Wright (1998) used a decision tree approach to UT classification, which enabled her to inspect the discriminative power of various prosodic parameters. She found that duration and F_0 were used on 88% of occasions, and that the corresponding figure for RMS energy was only 12%. Other writers, such as Abercrombie (1967), contend that loudness has little linguistic significance, in the sense that it does not contribute much to comprehensibility.

Speaking rate and rhythm are, of course, in the time domain. They reflect the relative durations of syllables and other sub-utterance units, so one of the generally accepted acoustic parameters of

prosody is **duration**. It is different in nature to energy and F_0 – as Hirst et al (2000) have it, there is no such thing as a "duration curve" along the lines of an energy or F_0 curve. Duration modelling depends on the units chosen for a given study, so that those investigating lexical stress, for example, will be primarily interested in syllable-level phenomena. Work such as that presented in this thesis, or Hirst et al's, partitions the utterance into convenient segments and computes a delta time as one of the parameters. Thus, durational analysis is a *sine qua non* of a segmentation approach. Length of utterance, syllables or other units, and such matters as pause location are not taken into account.

The PLoNQ analysis relies on three prosodic features or parameters of a segment: its F_0 gradient, F_0 midpoint and duration. It is important to note that *segment* here refers to a set of contiguous acoustic frames, whose length is determined by the segmentation algorithm described in Chapter 4, not a phone or other phonologically determined entity.

1.3.4 Functions of prosody

So far, we have looked mainly at examples where prosody aids lexical disambiguation: it reveals what a word or structure denotes. But in non-tone languages, prosody plays a more important role in connotation than in denotation. Couper-Kuhlen (1986:111) contrasts two spoken versions of the utterance *Shut the door*. A falling tone on the final word imbues the utterance with an abruptness which leads the listener to feel that an **order** is being issued, whereas a rising tone at the same location would cast the utterance more as a **request**.

A further example given by the same author, with an identical tonal contrast, is *John's going home*. With the falling tone, the utterance could be characterized as a **statement**, whereas the rising tone turns it into a **question**, or at least a request for confirmation or clarification. The objection could be raised at this point that the distinction is not the same as for *Shut the door* – the two articulations of the second utterance are semantically different, and it is not just a matter of connotation. To respond to this criticism, we can turn to **speech act theory**.

1.3.4.1 Speech act theory

According to Searle & Vanderveken (1985:8), an **elementary sentence**, which is one type of speech act, has the form *f(p)*, where *f* is an indicator of **illocutionary force** (IF) and *p* the **propositional** content. All utterances must have both components, except in a few marginal cases. The utterances *Shut the door!* and *You will shut the door* have the same propositional content, namely, that the door is supposed to be shut by the addressee, while the respective IFs are of command and of prediction. In these examples, the lexis and grammar, as well as pitch modulation, realize IF, while in the two examples of the previous paragraph, only pitch modulation has this function.

Searle & Vanderveken provide for a subclass of elementary sentences called **performative sentences**, such as *I promise that I will do the dishes* and *I pronounce you man and wife*. These utterances, it will be seen, are special in that the very act of making the declaration brings about the change in the world that is predicated on their propositional content: the promise or pronouncement is made, as it were, in the uttering. This is subject to certain constraints, of course, such as the qualification of the speaker of the second utterance to conduct a marriage. Performative verbs, incidentally, are often syntactically marked in English, in that the contrast that normally obtains between the simple present and present progressive tenses, to indicate respectively habitual activity and current activity, does not extend to verbs of this type.

Notice that any utterance can be rendered performatively if desired, stilted though the results may be; thus, *I declare that John's going home*, *I predict that you will shut the door*. Austin (1962) asserted that there are as many types of IF types as there are performative verbs, that is to say more than one thousand, distributed amongst five major categories. Searle (1976) disagreed with Austin's taxonomy on the grounds that there are IF types which are never realized performatively, such as *insinuate*. Another objection was that the inventory of IF type should be language-neutral (applicable to any natural language), and not constrained to the English lexicon.

Searle also proposes five major categories. There is no doubt that the bulk of what is uttered in ordinary language would be assigned to two of these categories, namely **representatives**, which include ordinary statements and assertions, and **directives**, a class which comprises requests, orders and questions, among others. There are no reports of experiments to test whether prosodic features could be pressed into service in disambiguating between these categories – whether, for example, better results could be achieved using Austin's or Searle's taxonomy. The fact that requests, orders, and all types of questions (wh-questions, yes/no questions and tag questions) are the most likely candidates for prosodic disambiguation, and that they all come under the category of directives, suggests that this would not be a promising avenue of research.

Before closing this section on speech acts, it is worth noting Searle's and Austin's **indirect** speech acts. Searle gives *Do you know the way to the Palace Hotel?* And *Sir, you're standing on my foot.* Here, what is intended by the speaker is a little different from what is expressed, such that the hearer should interpret the first example as a request for information, rather than a yes/no question, and the second a request for action, not a mere statement of fact. Some related issues arise from the annotation of the Oasis corpus, and they are mentioned in Chapter 3.

1.3.4.2 Attitudinal functions of prosody

Jensen et al (1994) comment that an important role of prosody is to convey the speaker's mood, relationship with the hearer and other attitudinal matters. In a pedagogical approach to prosody, O'Connor & Arnold (1961) set out over a hundred attitudinal categories, with associated pitch tunes. Among them were such finely differentiated items as "mild surprise but acceptance of the listener's premises", "critical surprise" and "affronted surprise"; their taxonomy is fine-grained to say the least. It was not designed for computational classification tasks, of course, but it does help us to understand how difficult it would be to characterize prosodic features in a rules-driven manner. It may be perfectly obvious to the layperson that prosody is capable of carrying anger, sadness, happiness and a host of other emotions, but practically impossible to pin these emotions down to specific prosodic patterns.

In the corpus chosen for these experiments, we can expect a reasonable amount of attitudinal variety. Some of the speakers are frustrated, some businesslike; many of them are chatty and friendly, and a few are drunk. A degree of familiarity with the data leads one to form associations between some of the UTs, as annotated in the corpus, and the mood of the speakers. Those calling to report a problem, for example, tend to demonstrate the fact in their tone of voice, those requesting action to be taken are frequently matter-of-fact and somewhat curt-sounding, and requests for information are generally friendly and light-hearted. Experiments with human subjects are reported in Chapter 6, and some of their subjective views on the mapping between tone of voice and UT are recorded there also.

It is on this mapping, it is considered, that the automatic UT classification will stand or fall. Attitudinal features pervade an entire utterance, and have a real hand in shaping its prosodic contour, whereas the key to the IF of an utterance (whether it is a question or command, and so on) resides in the final part of the utterance alone.

1.4 Prosodic transcription

One of the goals of this work was to design a system for automatic annotation of prosodic data. As has been explained, this was achieved via a linear regression segmentation of the F_0 contour, followed by clustering, to establish a discrete set of feature vectors, which would constitute the set of prosodic labels. An alternative approach, Hirst's MOMEL and INTSINT algorithms, is treated in Chapter 7, and two other transcription systems are described briefly here.

1.4.1 ToBI and Tilt

ToBI (Tones and Break Indices) transcription (described by Wightman (2002)) uses the notation of Pierrehumbert (1980). At the so-called **intonational** level, the symbols L and H are used to describe respectively low and high **pitch accents** (marked with the diacritic *) and **boundary tones** (marked by %). Thus, a high pitch accent would be marked by H*, with the diacritic marks describing pitch prominence and phrasing, and the letters giving an indication of overall pitch shape. A further diacritic ! indicates downstepping, Wightman explains, while a special label HiF0

is used to specify the highest pitched point of a major phrase. Where a boundary tone and pitch accent coincide, a + symbol is used.

This is obviously rather different from the annotation scheme presented in Chapter 4, which seeks only to provide the pitch shape information, and does not attempt to locate phonological events such as boundary tones. In general, the annotation work has to be performed by human experts, although there have been experiments with automating the process. Rapp (1998) achieved labelling accuracy of 78.7% by experimenting with optimal sets of duration, F_0 and energy features. In prosodic labelling of Japanese, Noguchi et al (1999) report a labelling precision of 58.1% and 68.6% for pitch accents, with recall markedly lower than that, in a procedure which incorporates morphological analysis and word alignment as well as F_0 extraction. Obviously this would not be a suitable approach for our purposes, because in this data-driven approach no word-level analysis is available.

Grabe et al (1998) point out that ToBI transcribers sometimes have difficulty reaching agreement on the right labels, because the inventory of tones is not sufficiently constrained. This is a criticism levelled also by Taylor (2000); he also notes that the distribution of labels is uneven, and that indeed in the ToBI annotation of the Boston Radio corpus 79% of events were labelled H*. Such an annotation, clearly, would be useless for any classification task on which it depended: it ignores differences between members of the large class, and precludes any probabilistic discrimination between it and other classes. Taylor's **Tilt** model of prosodic description, he claims, avoids this problem because it uses continuous parameters to describe a pitch contour, rather than the discrete classes of ToBI.

Taylor first uses a peak-picking algorithm to detect prosodic events – essentially the same pitch accents and boundary tones as ToBI. His next phase assigns Tilt parameters to the events. The parameters are *amplitude*, that is the peak F_0 value of the event; its *duration*; and the parameter *tilt* itself, a dimension-less value between -1 and 1 which indicates the proportion of fall and rise in the event.

Taylor argues forcefully for the use of continuous parameters in prosodic analysis, contrasting the position with that of segmental phonology where discrete labels are indeed valid. Thus, even though the distinction between realizations of /b/ and /p/, in English and many languages, lies at an unspecified position on a continuum, there is nevertheless a point, if voicing is gradually increased in an experimental situation, at which the native speaker/hearer can decide whether they have heard *bill* or *pill*. There is no concept, says Taylor, that is "a bit 'bill'-like and a bit 'pill'-like," whereas in the realm of prosody, one can imagine attitudes that are in some sense conflicting – joviality and sarcasm, say – being conflated, and yielding, for the hearer, a combined impression of the two.

This argument is specious. If the experiment Taylor mentions (without citation) draws the correct conclusion, then we must allow that people can always tell the difference between the two sounds. If it is wrong, we can surmise that people may sometimes be confused as to which sound is which; either way, the experiment is silent on any exotic semantic cocktails the auditory experience might or might not serve up. Furthermore, it *is* the case that discrete categories are required in prosody, even on Taylor's account, where there are lexical tones. If a foreign speaker of Mandarin pronounces *ma* with a tonal pattern lying between the first and second tones, the Chinese hearer will conclude that the foreigner's pronunciation is unclear, not that he is evoking some fabulous creature that is "a bit horse-like and a bit mother-like."

An alternative motivation for retaining discrete categories in segmental phonology, and eschewing them in prosody, could be the intention of the speaker. His pragmatic target, in the latter case, could well lie somewhere between two attitudes; or he could be seeking to convey both.

These matters are not pursued further here, because as it turns out we do require a discretization of the prosodic contour, in order to conduct the classification experiments. Our goal is to take prosodic labels, and sequences of these labels, to determine which ones best characterize particular classes of operator enquiry. With continuous parameters, this would simply not be possible; one would enjoy as much success if one tried to classify unseen data by the stylized training contour, or indeed the F_0 curve itself.

Tilt-based classification systems, such as Wright (2000), do accept variable parameters. Wright's work employs decision trees, whereby the values of a large number of experimenter-specified parameters, for example "Tilt amplitude of utterance-final event", are compared to thresholds generated by the tree technology.

1.5 Summary

This introductory chapter began by offering motivation for a classification system that is novel in several respects: it is data-driven, and requires very little manual intervention. The corpus used represents spontaneous speech, and it was built in a legitimate industrial setting. After setting out directions for the rest of the thesis, a number of terms used in the domain were pointed up: prosody, intonation, pitch and fundamental frequency. The decision to base the experimental work on pitch modulation alone, ignoring other candidate components of prosody, was explained.

The power of prosodic features to discriminate semantically was initially demonstrated by an account of lexical tone in language. The attitudinal functions of prosody were discussed, and some account was given of prosody as it relates to speech act theory. Finally, two well-known models of prosodic annotation were described, and it was explained why their use would not have been appropriate to the needs of this research.

2. Class assignment in language: a literature review

The last chapter was devoted to the theoretical background to prosodic analysis, and the relationships between it and speech act theory. This chapter deals with more practical matters, examining the work of others who have attempted to distinguish dialogue acts by examining linguistic features at different strata, including discourse context, vocabulary and prosodic features. It will be shown that this work has met with varying degrees of success; and that while some of it, like that presented in this thesis, is essentially data-driven, rules-based systems have an important part to play too.

Since PLoNQ is a system based on prosodic analysis, particular attention is paid to classifications based on that stratum, and its merits and limitations are discussed. First, though, we look at linguistic classification systems in general, and attempt to show that dialogue act typing is in fact a special case of the multi-class problem.

2.1 Linguistic classification tasks

In information retrieval (of which the commonest manifestation today is the search engine), a user enters keywords, and is offered a ranked list of possible solutions from a virtually unlimited set of possibilities. Keyword-based **topic identification**, as described for instance by Wright et al (1995) and Smith & Russell (2001), attempts to find the single best solution to a multi-class problem from a very constrained set of candidates, or alternatively to rank each member of that set.

The principle is straightforward: examine a training set, and attempt to ascertain which features (in this case keywords, or perhaps sequences thereof) best characterize each topic. Next, find out which features are best represented in some unseen text, and conclude that it belongs to the class, or topic, characterized by those features. Experiments in topic identification will be discussed at 2.3 below; for now, let us simply note that it is, in a way, the antithesis of IR. The first seeks to determine the right topic given a document, while the second looks for documents on a stated topic.

Other classification tasks in language processing are now considered.

2.2 Automatic language identification (LID)

LID is an interesting task which also boils down to the assignment of a text to one member of a closed set. One can imagine a machine translation system which is capable of handling various language pairs: a LID module would spare the user the step of specifying the source language. An alternative application, this time for spoken LID, might be a multilingual call centre, where callers could be automatically routed to the operator fluent in the required language (Lazzari, Frederking & Minker, 1999).

Dunning (1994) shows that "the algorithm used can be derived from basic statistical principles [and] is not based on hand-coded linguistic knowledge, but can learn from training data". He rejects rules-based systems relying on strings of characters unique to a particular language, or lists of common words (citing that proposed by Johnson (1993)) on the grounds that these strings would be unlikely to occur in short texts. For his own probabilistic model, he claims classification accuracy between 92% (20 bytes of test text and 99% (500 bytes), with a training set of 50Kb. Dunning uses Markov models to determine the probability of an observed sequence of characters.

Dunning also applied the technique to tackle an analogous classification task in an entirely different domain, successfully identifying types of organism (human, E Coli or yeast) from DNA sequences.

Thymé-Gobbel & Hutchins (1996) describe a spoken LID system called **Discrim**, which attempts to discriminate between languages on a pairwise basis. Their system, like PLoNQ, relies on input from prosodic features. It was noted in Chapter 1 that PLoNQ can only offer limited performance, and that its real potential would only emerge if it was used in combination with an analysis from other linguistic strata. In similar vein, Thymé-Gobbel & Hutchins characterize Discrim as "only a component of a complete LID system incorporating … phonetic events and word recognition". Nevertheless, they conclude that the system discriminates effectively between some language pairs. Their best results are for the Mandarin-Spanish pair: a consequence, they maintain, of the very different pitch patterns of these two languages, "the overall flat pitch of Spanish" contrasting well with the tonal system of Mandarin.

Discrim extracts pitch and amplitude information and performs a syllable segmentation, assigning to each features such as syllable duration and pitch difference between current and next syllables. There are 244 feature types in all, but only a subset is used, the discriminatory features being computed by log-likelihood ratio functions. In 2.5, we shall see that in other work (Jurafsky et al 1997), a different method – classification and regression trees, **CART** – is used to decide feature salience.

2.3 Topic identification experiments

As an initial experiment in multi-class problem solving, we built a baseline **topic-identification** system, implementing algorithms of (and effectively replicating the work of) Garner (1997) and Wright et al (1995). This work was reported in Smith & Russell (2001).

The aim was to build a system which would correctly identify the topic of an unseen news text, by examining the frequencies of significant keywords used. To do this, we first decided on ten appropriate topics: domestic politics, foreign politics, health, finance, crime, sport, media,

technology, environment and education. Two sets of training data were assigned to each category, one set consisting of eight news stories, the other – a subset of the first – of three stories. All the texts were taken from BBC Online News, where they are arranged in categories not dissimilar to ours; the fact that multiple category membership of texts is commonplace in the BBC scheme is what led us to develop our own. Often, though, the topic of a text seemed to motivate its assignment to more than one class (a story about Elizabeth Taylor and her AIDS-related work belonged intuitively to both media and health categories, for example). In such cases, we simply excluded the text from training data.

Once training data had been prepared, the probable topic of an unseen text was computed by maximizing probability (2.1) (Garner 1997) over all keywords and all topics:

(2.1) $P(x|m_i, D) = P(w_1|m_i, D)P(w_2|m_i, D)\ldots P(w_K|m_i, D)$

Here, $P(x|m_i,D)$ represents the probability that an observation x, instantiated as the unseen text, will occur in the training data D associated with a topic m_i. Whenever a word encountered in the unseen text is found in the training data for a particular topic, its probability of occurrence ($P(w_j|m_i,D)$) is computed by dividing the number of tokens of that word in topic m_i by the total number of words in D; if D does not attest the word in question, the probability defaults to an arbitrarily small value, the floor value, which is lower than the probability estimated for any observed word. As (2.1) shows, these probabilities are then multiplied together to yield $P(x|m_i,D)$. Normally, $P(x|m_i,D)$ would in turn be multiplied by the topic prior probability before maximization, but that step is skipped in this implementation, as it is assumed that all topics are equally likely.

When the small (three stories per topic) training set was used, 30 out of 50 stories were assigned to the right topic, while 42 stories were correctly matched by means of the eight-story training set.

2.3.1 Data thresholding

Various information-theoretic **data thresholding** measures, such as **usefulness** (Wright et al 1995), **salience** (Gorin 1995) or **mutual information** (MI) are described in the literature. They aim to identify structure which is, in some sense, optimal for discriminating between the different classes in question. These techniques are most commonly applied at the word level, either to text (as in the case of our topic identification experiment), verbatim transcriptions of speech, or the output of a speech recognition system. In Chapter 4, we motivate and discuss the inclusion of two of these measures in our utterance classifier.

Our topic identification program, therefore, next applied (2.2) (Wright et al 1995) to determine a usefulness score for each vocabulary item in the training data.

$$U_k = P(w_k|T)\log\frac{P(w_k|T)}{P(w_k|\overline{T})} \quad (2.2)$$

A word *w* is thereby said to be useful when it is frequent in training texts of topic *T*, and occurs relatively rarely in other texts; the usefulness score describes the discriminatory contribution of keywords to the topic of the text. Thresholding or pruning is then carried out so that only the *n* most useful keywords are searched for when determining the topic of an unseen text.

Table 2.1 shows, for each news topic, what were computed by the usefulness algorithm to be the top ten keywords. The lists contain a reasonable proportion of words which could be said to characterize each topic.

Table 2.1 Top ten keywords in BBC news corpus ranked according to usefulness

finance	*computers*	*crime*	*domestic politics*	*education*	*environment*	*foreign politics*	*health*	*media*	*sport*
banks	internet	plane	party	students	environment	eritrea	cancer	film	boxing
yen	web	victim	tories	schools	masts	lazio	tamoxifen	olsen	sydney
merger	websites	suharto	kennedy	college	gm	ethiopia	breast	laurel	olympic
bank	engines	bomb	labour	oxford	fuel	eritrean	parodi	travolta	ham
jp	lottery	victims	mp	university	crops	speight	fruit	films	spurs
shares	computer	cheng	ira	pupils	we	monitors	everson	magazine	garcia
sega	information	police	donaldson	curriculum	oil	kosovo	boots	hardy	talent
debenhams	sites	trial	ulster	comprehensive	pioneer	fiji	krishnamurthy	hurley	robson
chase	neurons	musharraf	trimble	state	environmental	ethiopian	removed	comedy	mcgrath
banking	identity	li	romsey	i	prince	electoral	pacemaker	book	edwards

However, when topics were assigned to unseen (test) news stories, greater accuracy was on the whole achieved when all training data was considered than when thresholding was applied. Table 2.2 shows how many of 50 stories were correctly identified when usefulness thresholding was applied at various levels *n*: that is to say, when only the *n* most useful words in the training corpus for each topic were taken into account.

Table 2.2 Topic identification performance, showing the number of stories correctly classified out of a total of 50, using three different thresholding algorithms, trained on 8 stories

threshold	*usefulness*	*modified usefulness*	*salience*
10	10	30	10
20	13	31	10
30	12	35	15
40	12	35	19
50	13	37	18
60	12	39	24
70	11	38	28
80	12	35	30
90	13	34	30
100	13	34	29
150	12	35	31
200	16	36	34
249	16	37	35
299	16	37	36
349	16	38	36
399	16	42	42
449	16	39	42
499	19	41	43
549	22	41	42
599	23	41	40
no threshold	42	42	42

Table 2.2 also shows the results of thresholding under a reformulation of the usefulness calculation which we term **modified usefulness**, suggested by Garner (1997). There is a marked improvement here, although the best performance is still achieved when no thresholding is carried out. The reformulation, shown at (2.3) differs only from (2.2) in that the absolute probability of occurrence of a word is ignored.

$$MU_k = \log \frac{P(w_k|T)}{P(w_k|\overline{T})} \quad (2.3)$$

The **salience** results are discussed in 2.3.2.

Part of the discrepancy between the results from the two formulations of usefulness arises because the original form tends to rank function words unduly highly, which leads the application to treat these items as keywords. Table 2.3 shows the ranking of the determiner *the* for each training corpus, under both formulations of usefulness.

The better performance without absolute frequency is not entirely surprising. With very small training and test sets, a rich pattern of meaningful keywords is unlikely to arise, and the most common tokens will naturally be function words.

Table 2.3 Ranking of *the*, a typical function word, under the two usefulness formulations

corpus	*modified*	*original*
computer	931	1032
crime	1026	1132
domestic	896	32
education	946	146
environment	1101	172
finance	912	1012
foreign	928	29
health	1038	518
media	951	1046
sport	1003	957

2.3.2 Salience thresholding

Gorin (1995) defines salience as "an information-theoretic measure of how meaningful a word is for a particular device [i.e. task]". Gorin's work focused on unconstrained speech driven routing of telephone calls to one of a number of human operators, each dealing with one task, such as reverse-charge calls, credit card calls and directory enquiries. Salience of a word to a class *T* is computed by (2.4).

$$sal(w) = P(T|w)\log\frac{P(T|w)}{P(T)} \tag{2.4}$$

P(*T*) is ignored for our purposes, as all topics are equally likely. We do not know the probability of the class given the word, but it can be derived from Bayes' Law, as shown at (2.5).

$$P(T|w) = \frac{P(w|T)P(T)}{P(w)} \tag{2.5}$$

P(*T*) is again ignored. *P*(*w*) is the number of tokens of a particular word in the whole of the training data (in principle divided by the total number of tokens of any word, but this is a constant); *P*(*w*|*T*) is the number of tokens of the given word divided by the total word count for the topic, as per the usefulness calculation.

It will be seen that whereas usefulness compares the likelihood of a token in on-topic and off-topic texts, salience relates incidence in on-topic texts and all texts, whether on- or off-topic. (2.4) can be rewritten as (2.6), pointing up the relationship between usefulness and salience.

$$\begin{aligned} sal(w) &= P(T|w)\log\frac{P(w|T)}{P(w)} \\ &= \frac{P(T)}{P(w)}U(w) \end{aligned} \tag{2.6}$$

As Table 2.2 shows, at low pruning thresholds, the algorithm performs as badly as the original

usefulness formulation, but the performance accelerates towards that of the modified usefulness.

2.4 Dialogue act recognition

Garner (1997) and Bird et al (1995) established a clear link between topic identification and dialogue act recognition, applying maximum likelihood techniques from the former task to the latter. A stated aim of Garner's paper is "to formalize the theory used for topic identification such that it can be applied to dialogue move recognition in a robust manner".

Garner applied a maximum likelihood model to determine dialogue move in the Edinburgh Map Task, a corpus which incorporates dialogue act mark-up.

He attained an accuracy of 47%, which does not sound particularly promising until it is recognized that his model treats both moves and keywords as independent, that is to say no account is taken of contextual information. Furthermore, the approach is based purely on lexical frequency, and no appeal is made to other linguistic strata.

In this work, Garner also took advantage of and compared various data thresholding techniques, including usefulness, salience and MI. Figure 2.1, taken from Garner & Hemsworth (1997), shows very similar results to those of our topic identification experiment, set out in Table 2.2. Recall from 2.3.1 that Garner's formulation of usefulness is referred to as "modified usefulness" in the table; their performance under usefulness thresholding, like ours, is quite high even for a fairly low dictionary size (that is to say a high threshold, admitting only the most significant keywords), and climbs slowly towards the peak score, where the vocabulary is not pruned.

With salience thresholding, the two sets of results (Garner & Hemsworth and Smith & Russell) are again comparable. Performance is worse than usefulness with a low threshold, but gradually accelerates towards the peak.

The lack of success with the thresholding measures is attributed to data sparseness. Potentially significant keywords simply do not occur often enough to be true indicators of usefulness or salience.

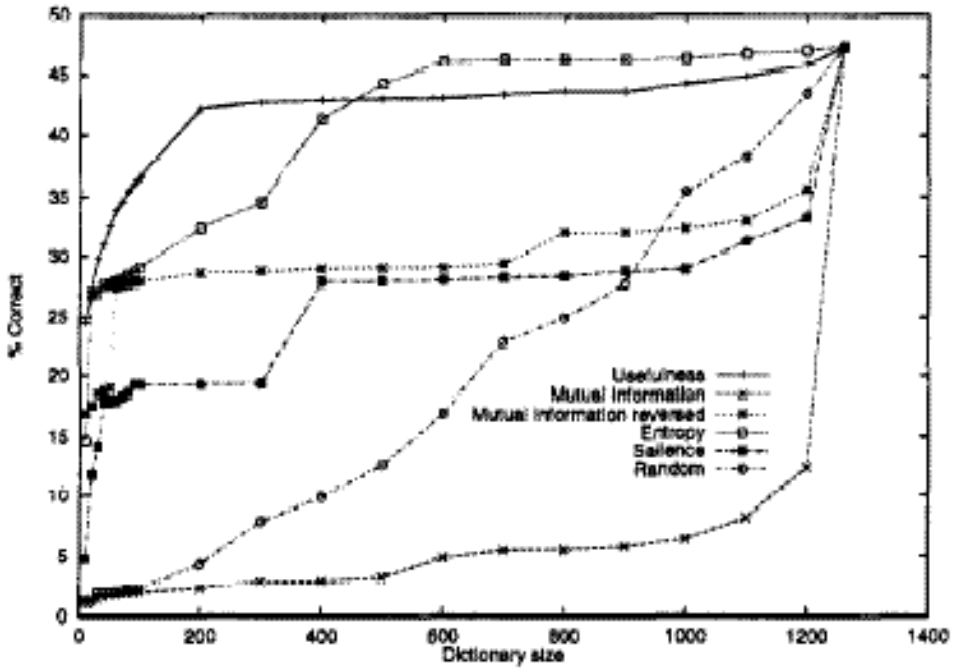

Figure 2.1 Comparative performance of various thresholding approaches, at different thresholds (dictionary sizes) by Garner & Hemsworth (1997)

Samuel et al (1998) used a **transformation-based learning** (TBL) heuristic algorithm to assign UT to texts. They derived a set of ordered rules, by hand-crafting rule templates such as (2.7).

(2.7) IF utterance u contains the keyword w

THEN change u's tag to Y

Keywords, or **dialogue act cues**, were selected where the entropy of their associated UTs was below a given threshold (Garner (1997) also experimented with entropy as a thresholding mechanism).

Samuel et al (1998) next generated all the **potential rules** that would render at least one UT in their training data correct. After initializing all utterances in the corpus as *suggest*, they associated with each potential rule an **improvement score** (the difference between the number of correct UT labels in the corpus with and without the contribution of the rule). The rule with the greatest improvement score was applied, modifying the UT on a number of utterances, and learned by the model; second and subsequent passes involved recalculating the improvement scores and executing the UT transformations again, until the greatest improvement score failed to attain a preset threshold.

In their data, over 71% of utterances were correctly tagged by this means. In a Swedish laboratory, Lager & Zinovjeva (1999) applied the TBL algorithm to the Map Task, attaining an accuracy of 62.1%.

2.4.1 Dialogue act recognition from discourse context

Nagata & Morimoto (1993) trained a corpus of conference-booking dialogues using a trigram model of utterances classified by speech act type, and attempted to predict, using mutual information, the following utterance type given the current one. As with some of the prosodic analyses described in 2.5, their main goal was to reduce errors in speech recognition.

Their approach is entirely probabilistic, and ignores any linguistic or lexical considerations: they report a classification accuracy of 61%, amongst the nine speech act types catered for. These include, for example, *phatic* (such as greetings), *request*, *inform*, *questionif* (yes/no questions) and *questionref* (wh- questions). There is also a special type, *DBM*, which denotes the beginning or end of a dialogue. A real dialogue, that is a succession of utterances, is modelled as a sequence of speech act types $s = s_1,...,s_n$ (where s_i represents the ith speech act in the sequence). In (2.8), the right hand term denotes the product of probabilities of some speech act type, given the speech act type that preceded it.

(2.8) $$P(s) = \prod_{i=1}^{n} P(s_i | s_{i-N+1}, \cdots, s_{i-1})$$

Note that the *i*th speech act is assumed to depend only on the previous *N* speech acts.

As we shall see in the next chapter, the corpus chosen for our utterance classifier consists only of isolated queries. In our experiments, therefore, no appeal could be made to discourse context.

Jurafsky et al (1997) labelled a portion of the Switchboard telephone corpus with 42 different utterance types. They combined discourse grammar (estimation of type based on adjacent utterances, as with Nagata & Morimoto) with keyword-based classification, and claimed accuracy of 64.6% (compared to 42.8% for their own keyword-only classification) in assigning utterance types to a test set. Table 2.4 shows the most common utterance types in their corpus, with examples.

Table 2.4 Utterance types in the Switchboard telephone corpus, with examples and incidence of each type

Tag	Example	Count	%
Statement	*Me, I'm in the legal department.*	72,824	36%
Backchannel	*Uh-huh.*	37,096	19%
Opinion	*I think it's great*	25,197	13%
Agree/Accept	*That's exactly it.*	10,820	5%
Abandoned/Turn-Exit	*So, -/*	10,569	5%
Appreciation	*I can imagine.*	4,633	2%
Yes-No-Question	*Do you have to have any special training*	4,624	2%
Non-verbal	*<Laughter>, <Throat_clearing>*	3,548	2%
Yes answers	*Yes.*	2,934	1%
Conventional-closing	*Well, it's been nice talking to you.*	2,486	1%
Uninterpretable	*But, uh, yeah*	2,158	1%
Wh-Question	*Well, how old are you?*	1,911	1%
No answers	*No.*	1,340	1%
Response Ack	*Oh, okay.*	1,277	1%
Hedge	*I don't know if I'm making any sense or not.*	1,182	1%
Declarative Question	*So you can afford to get a house?*	1,174	1%
Other	*Well give me a break, you know.*	1,074	1%
Backchannel-Question	*Is that right?*	1,019	1%

2.5 Evidence from prosodic features

2.5.1 Using decision trees

Jurafsky et al also took into account information from another linguistic stratum in their classification: namely, prosodic features of the speech stream. They set up several dozen utterance-wide prosodic feature types, including *f0_max_utt* (the maximum fundamental frequency reached), *rel_nrg_diff* (ratio of RMS energy of final and penultimate phrasing region), and *mean_enr_utt* (mean speaking rate value); then they trained **classification and regression decision trees** (CART) (Breiman et al 1984) whose task was to distinguish between two utterance types. The path through the tree, and eventual classification, was achieved through comparison of the feature values to constants stipulated at decision points.

The CART tree-building technique employs **binary recursive partitioning**. Given training data (speech segments, the dialogue act types they represent and the prosodic feature values associated with them), configurations of parent nodes with two children are hypothesized. Each bifurcation represents a decision point on a particular feature; the algorithm examines all possible values for that feature, and attempts to find a **splitting rule** that maximizes the discriminatory power of the feature. Thus, while the choice of features is made by the experimenters, the trees are derived by data-intensive means.

The tree shown in Figure 2.2 is taken from Shriberg et al (1998). It discriminates between the *backchannel* and *agree/accept* dialogue acts — two classes which the keyword approach often failed to distinguish.

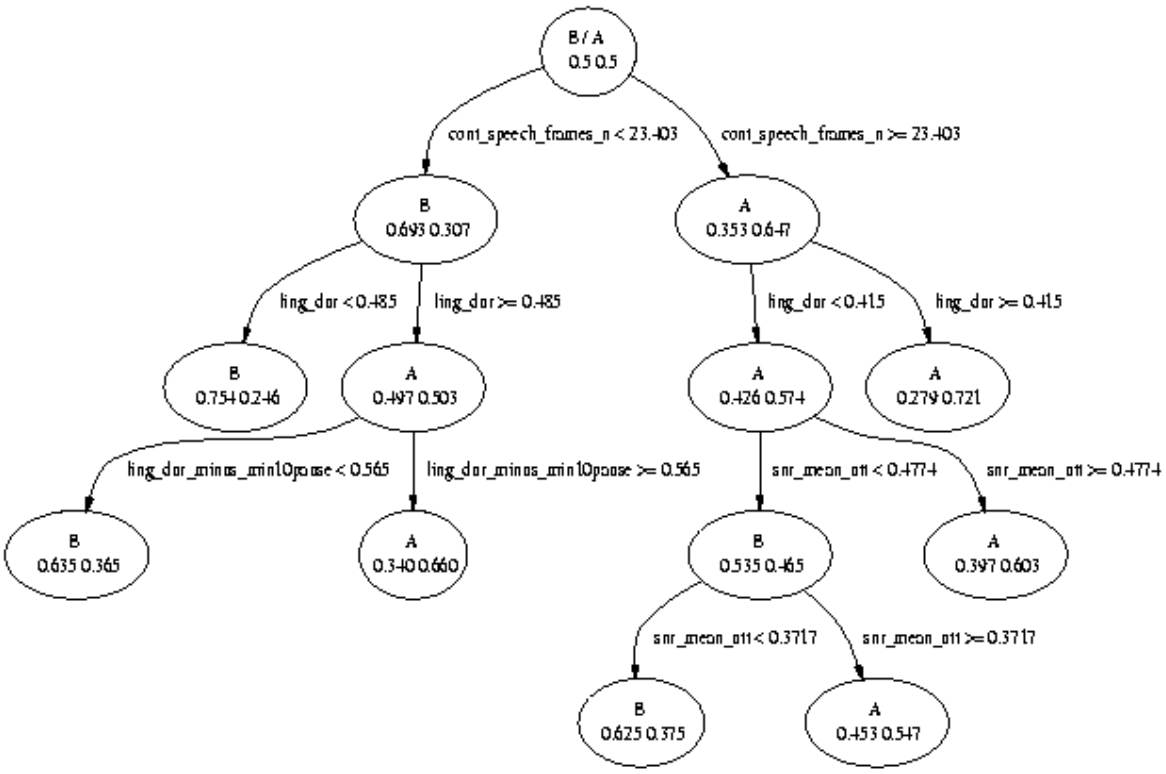

Figure 2.2 Decision tree distinguishing *backchannel* and *agree/accept* dialogue acts in the Map Task corpus, showing the bifurcation at decision points

2.5.2 A comparative study

Wright (1998) used prosodic features to determine dialogue move type in a Map Task corpus. Her goal was to reduce word error rate in a speech recognition system. She compared three approaches – HMM, CART and neural nets – to the modelling of prosodic features, first annotating speech according to the Tilt labelling scheme, which, as explained in 1.4.1, sets four Tilt parameters for each intonational event. Wright's HMM approach is essentially data-driven: although the Tilt labelling was trained on manual annotation, only the Tilt parameters are passed to HMMs as observations. The CART and NN approaches, however, exploited 54 carefully chosen prosodic feature types, in line with Jurafsky et al (1997). Results from the three approaches scarcely differed, despite the feature poverty in the HMM case. This last yielded 42%

correct classifications, with NNs reaching 43% and CART 44%. When Wright incorporated some discourse-contextual information, performance by the three approaches improved to 64%, 62% and 63% respectively. Chance (the proportion of moves actually belonging to the largest class) was 25%.

2.5.3 VERBMOBIL experiments

Nöth et al (1999) report experiments using neural nets to establish "the salient regions of a phrase" from pitch information in the VERBMOBIL database, a corpus created to train a spoken machine translation system. The corpus is annotated by UTs such as *suggest* and *accept*. The group found that those keywords which might be expected to capture the sense of some UT (such as "suggest" or "idea" in the case of *suggest*) are more likely to have this effect when they are stressed. The trajectory of Figure 2.3, taken from Nöth et al, is an impressionistic indicator of stress probability; it does not represent a stylization of pitch.

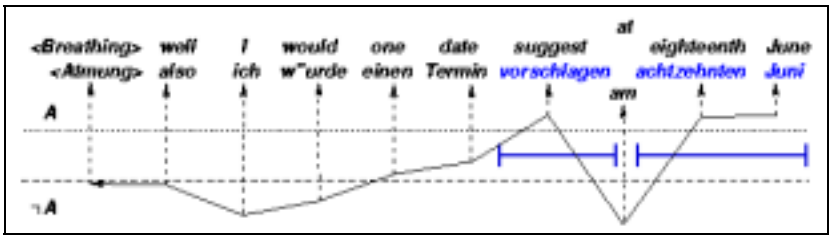

Figure 2.3 A VERBMOBIL sentence with word stress probability, showing "the salient regions of a phrase"

There is a body of related work exploiting prosodic features in speech recognition, including that of King (1998), who integrates language model, dialogue context and intonational information to improve recognition of spontaneous dialogue speech. Jensen et al (1994) present a scheme for phrase-level recognition of intonation contours, and show how it can help compute the perceived pitch of voiceless utterance segments (where no fundamental frequency measurement is available). Hirschberg et al (1999) found, like Wright, that prosodic features can be used to predict recognition errors: their work was based on dialogues from the TOOT train information corpus. Carey et al (1996), working on speaker identification, established that prosodic features are less

sensitive to noise distortion than cepstral coefficients.

2.6 Concluding remarks, and a caveat

In this chapter, a literature review of work tackling linguistic classification issues was presented. We looked at the language identification, topic identification and UT recognition tasks, drawing a close parallel between the last two. As well as a survey of the work of others, some account of our own topic identification experiments was offered. A number of different algorithmic approaches were considered, as was the question of classification by analysis at various linguistic strata.

This thesis is chiefly concerned with analysis at the prosodic stratum, of course. Now, despite the intuitions linking speaker intent with prosodic realization mentioned in Chapter 1, it must be understood that any production software or marketable device would almost certainly draw on knowledge sources at other levels. Recall that Thymé-Gobbel & Hutchins (1996) intended Discrim to be just a module of a larger LID system, with access to segmental information also. Jurafsky et al (1997) reported UT classification accuracy of only 38.9% by prosody alone, but this figure increased to 65% when combined with recourse to discourse context and word recognition. Nöth et al (1999), for their part, hold that while " 'pure' prosody can be used to recognize accentuation or prosodic boundaries … afterwards, however, this pure prosody approach has to be combined with word information."

It is, in fact, possible to attain reasonable results on the basis of prosody alone; the experiments reported in later chapters, as well as the work summarized in the present chapter, bear testimony to this. Any performance significantly better than chance constitutes evidence that prosody, a traditionally somewhat neglected knowledge source, has an important discriminative role to play in UT detection and other linguistic classification tasks.

In this chapter, reference has been made, in passing, to a number of spoken corpora used for prosodic research: the Map Task, for example, is particularly popular in the field. This is the topic to which we now turn our attention.

3. Data: choosing and using the corpus

The selection of an appropriate corpus of spoken English was an important step in this research. Earlier work on dialogue act classification was reviewed in the last chapter: the corpora used by these researchers were potential candidates, although the ATR corpus used by Nagata & Morimoto (1993) was in any event not available. The Switchboard (Godfrey et al 1992) and the Map Task (Andersen et al 1991) corpora, used in the work of Jurafsky et al (1997) and Garner (1997) respectively, were available. A clear advantage of using one of these corpora would have been that the earlier findings could have provided useful benchmarks for the results presented here.

However, an important aspect of the present research was to establish how useful linguistic features can be in determining query types. For this reason, a corpus of spoken queries was sought, as opposed to a collection of connected dialogues, such as the Map Task or Switchboard.

3.1 The ATIS corpus

The Air Travel Information System (ATIS) is described by Hemphill et al (1990). The corpus was compiled as part of a Wizard of Oz experiment, in which participants were assigned air travel planning tasks such as (3.1), given by Hemphill et al.

(3.1) Plan the travel arrangements for a small family reunion. First pick a city where the get-together will be held. From 3 different cities (of your choice), find travel arrangements that are suitable for the family members who typify the "economy", "high class", and "adventurous" life styles.

The participants were asked to plan the arrangements by speaking natural language queries regarding timetables, fares and so on to the system. The system, in fact, consisted of an individual consulting a publication called the "Official Airline Guide", but this was not revealed until the end of the task. The queries were recorded and transcribed, and those that met certain criteria (such as context independence, clarity and well-formedness) were assembled into a corpus which is available from the Linguistic Data Consortium. The corpus is lemmatized and POS-tagged, but no other mark-up – such as type of query – is available.

For use in the work of this thesis, the ATIS corpus would have to have been annotated with query type information. This was not done, as it was deemed too time-consuming; but it is worth pausing to consider ways in which the task could have been performed. A look at the ATIS queries in the first column of Table 3.1 (extracted from Linguistic Data Consortium 1993) might lead one to put forward the five categories shown in the second column, as potentially useful topic-orientated classes. A characterization of each query with respect to the type of response expected, as in the third column, would perhaps correspond more closely to speech act theory, as discussed in Chapter 1; intuitively, too, it would yield better results on an analysis based on prosodic features.

Table 3.1 Actual queries from ATIS corpus; and two putative utterance classification schemes

Query	Topic	Response type
Show me all the nonstop flights from Dallas to Denver early in the morning.	times	list
What airline is CO?	codes	what
Show me all the nonstop flights from Denver to San Francisco, leaving about three o'clock in the afternoon.	times	list
Show me the distance from San Francisco airport to downtown	airports	what
Show me the airfares on flights from DFW to Denver before 9:00 A.M.	fares	list
What is restriction A P slash eighty?	codes	what
What is restriction V U slash one?	codes	what
What is the fare on flight eleven forty-nine from Continental Airlines?	fares	what
What is class Y?	codes	what
What type of aircraft is flying United Airlines flight nine fifty-three?	flight-details	what
Show me flight nine fifty-three 's arrival time and what type of meal it has.	flight-details	what
Is there more than one airport in the Boston area that American and Delta service?	airports	yes/no
What is a Y class , and what does the DL under FA column mean?	codes	what
Describe each of the different classes for airfares.	fares	list
What type of aircraft is flight number five four seven ?	flight-details	what

Show me the distance from the Denver airport to downtown.	airports	what

If the topic-orientated classes are too task-specific to be of use, the response-type classification presents drawbacks too. There are only three classes; most of the queries are classified as *what*; and the scheme is rather contrived inasmuch as it is difficult to imagine an application where the classes could be usefully exploited.

Perhaps part of the problem is that ATIS is – by design – purely an information-giving system. One can, however, conceive of a speech-driven system which not only responds to requests for information, but also allows the user to book and purchase tickets, and even handles statements of problems, such as "I can't get to the airport before six in the morning." Such a system might allow for a set of query types such as *information*, *action*, and *problem*: a linguistically motivated set, which could plausibly provide a useful discrimination function in some future live application. A corpus of spontaneous speech, annotated in this way, would serve the purposes of this research well.

BTexaCT's Oasis is such a corpus.

3.2 BT telephone enquiry corpora

BTexaCT, the research and development division of British Telecom, have compiled three corpora of operator service calls from the general public. One of these, known as the XML corpus because the annotation was carried out according to XML standards, includes transcriptions of entire calls to the operator, which may consist of a number of conversational turns by both parties. Unfortunately, speech files are not provided with this corpus, so it could not be used for a prosodic analysis.

With the other two corpora, the recorded data is supplied. They are known as **first utterance corpora**, because only the first conversational turn from the caller is recorded and transcribed: the only exception to this is when the first utterance consists solely of "Hello?" or a similar greeting, in which case the caller's second utterance is also included in the corpus. The **150 corpus** consists of calls to BT Customer Service, during the working day, including requests for new telephone lines,

enquiries about billing, complaints and so forth.

The Oasis corpus is similar, but it comprises calls made to the general operator by dialling 100. In this research, it was decided to use Oasis, for several reasons. This corpus contained a greater variety of call types, ranging from requests for alarm calls or connection to other parts of BT, to people asking for the date or time, or making requests which the operator could not be expected to fulfil. Oasis recordings were made at various times of the day and night, and it was felt that this would provide a broader sample of enquiries and enquirers for the study. Finally, while the 100 operator service is always answered by a human agent, the 150 service normally connects to a recorded list of options; however, for the purposes of the corpus compilation, calls were answered directly by an agent, and not under the conditions normally associated with the 150 service. Indeed, the corpus has a special annotation category for callers expressing surprise at being answered by a human being.

For each of the 8441 calls in Oasis, a sound file is provided, along with an annotated file description, as illustrated in Corpus Excerpt 3.1.

```
Oasis.3.tdef:phase5/day08/aa150020,B,,,phase5/day08/aa150020.1.msg,,info,0,ah
morning er ~ what's the k... # can you give me the ah oh put it this way and
start again #! <laugh> er @@ i want to ring {business}Notts {business}County
{business}Council at {town}Nottingham (right) i want the {business}Social
{business}Services department @@,dq,,,,,5,,,2,1.969,16.179,ok,good
```

Corpus Excerpt 3.1

In the description, fields are delimited by commas. Following the **unique call reference** and **sound file path specification**, the string *info* denotes the **primary move type**: in this case the annotator has determined that the caller's main purpose is to seek information. Available call classes and their distribution are discussed in 3.2.1 below. The 0 of the next field indicates that this is the first utterance of the caller, and was not preceded by a first utterance that consisted of a greeting only. After the transcription, the string *dq*, meaning "directory enquiries", identifies the **semantics** or **call class** of the utterance: this caller should have dialled 192 rather than 100, and

the operator probably next offers to transfer him to that service. The number 5 gives a measure of the quality of the transcription, on a scale from 1 to 5, and the 2 indicates that this is the second turn in the conversation, since the operator spoke first. The utterance began 1.969 seconds after the start of the call, and ended at 16.179 seconds. *ok* denotes the articulation quality of the speaker, which may also be represented by *poor*, and the final *good* describes the channel type, which can also be *bad* or *mobile*.

A good many of the fields are not relevant to this corpus annotation and are therefore left blank. This is because the transcription file format was designed, by BT, for use with other applications as well as this one. The second field has no discriminating function, as it is represented by *B* throughout the corpus, although BTexaCT (2001) describes it as the **speaker reference**. One of the fields, which is blank throughout Oasis, is a code indicating the status of the speaker as adult male, adult female or child. It is regrettable that this information was not available, as it would have been quite useful for the pitch extraction procedure described in Chapter 4.

In the transcription itself, some special conventions are used; they may be observed in Corpus Excerpt 3.1. The symbol @@ is used parenthetically to delimit the primary move of the utterance. It can be seen from the example that the speaker hesitates at first, and takes a little time to come to the main thrust of the utterance, which is marked using this symbol. The primary move type of the utterance, in this case *info*, relates solely to the bracketed portion. Pauses are represented by #, and non-speech sounds by #!. Comments, which may include a description of such sounds, are enclosed in angle brackets, and some characterization of proper nouns, such as the fact that Nottingham is a town, is given in curly brackets. Ordinary round brackets are used for backchannel speech from the operator, and occasional comments that may be overheard from third parties in the background.

In most of the calls, the caller first greets the operator or says something along the lines of “I wonder if you can help me”, before stating their main business. Sometimes, as noted above, the conversational turn consists only of a greeting, and in such cases the caller’s second utterance is also treated. Where the caller greets the operator and then moves on to other matters without waiting for the operator’s response, however, the opening part of the utterance is defined as a

social move; its end is indicated by a tilde symbol. In the transcription at Corpus Excerpt 3.1, the caller begins with “Ah morning”. The following filled pause, transcribed as “er”, is not strictly part of the social move, as BTexaCT (2001) points out; but where such items occur at the beginning of a move, they are shown as belonging to the end of the previous move “for historical reasons”.

3.2.1 Call type annotation

Call type annotation, then, was carried out at two levels: the primary move type, which “is intended to capture the sense of what the caller is trying to achieve” (BTexaCT 2001), and the call class, which is much more application-specific, often conveying the exact type of BT service required, such as directory enquiries, an alarm call or ADC (advice of duration and charge). Altogether there are 81 call classes, under 24 general headings such as “directory enquiries” and “alarm call”. The call classes available under “directory enquiries” are listed in Table 3.2, which was taken from BTexaCT (2001).

Table 3.2 Call classes related to directory enquiries in Oasis corpus

dq	Request for connection to directory enquiries
	Request for directory enquiries phone number
	Request for a phone number that requires directory enquiries
dq-area-inf	Information on an area code (what is it?, is it a mobile?, etc.)
dq-area-loc	Location of an area code
dq-area-num	Area code for a location
dq-area-old	Out of date numbers (includes big number changes)
dq-conf	Asks for confirmation that this is directory enquiries
dq-dead	Problems getting through to directory enquiries
dq-loc	Reverse directory enquiries
dq-tariff	Cost of directory enquiries

Some of these classes describe services that the 100 operator is unable to provide (and, in the case of dq-loc, cannot be provided by the directory enquiry operator either). In most cases, the 100 operator's action will be to redirect the call to the 192 service, or advise the caller of this number. Nevertheless, it is the service that the caller asks for which is noted in the corpus, regardless of its actual availability.

It would probably be extremely useful, in a call-routing task for example, to be able to predict, automatically, the call class to which a particular call should be directed: this is the function of the "How may I help you" application reported in Gorin et al (1995), and described in Chapter 2. Such a classification would require a probabilistic treatment at the lexical or phonetic level, however. The prosodic analysis which the present work relies on would not be sufficient to distinguish between requests for an alarm call and ADC, for example, so no such experiment has been attempted.

Intuitively, the primary move type annotation lends itself well to a prosodic analysis. The types, listed in Table 3.3 (BTexaCT 2001), correspond reasonably closely to those treated in the classification work of others, such as Nagata & Morimoto (1993), Garner (1997) and Jurafsky et al (1997), reviewed in Chapter 2, and to the categories of speech act theorists such as Searle (1976). One would not be surprised to find distinct differences in the realization, in terms of prosodic features, of a problem statement and a request for action, for instance. However, the *who* and *info* types represent the same speech act (as do *connect* and *action*), so one would expect a prosodic model to perform badly in making these distinctions.

Table 3.3 Instructions for annotation of primary move type of Oasis utterances

Primary move type	Question. Ask from top to bottom. Stop when you can answer 'yes'
Prob	Is there ***only*** a description of a *problem* or situation?
Who	Is it a request about *who* to contact. (e.g. which BT contact point or number to call?)
Info	Is it a request for *information or advice* (e.g. about BT services, number or account information, the state of the network, general knowledge, or time)?
Connect	Is it a request to be *connected* to another agent, service, person or organisation?
Action	Is it a request for operator *action* (e.g. named service; change to BT records or customer service options; initiation of a BT process such as line test; report a fault)
Other	Everything else

Table 3.3, showing the six move types, provides guidance for those annotating the corpus, giving examples of the correct annotation for particular requests, and the order of precedence for assigning classes.

The different move types in the corpus are not equally distributed. Figure 3.1 shows that *info* calls are more common than any other variety, while *connect* is the smallest category.

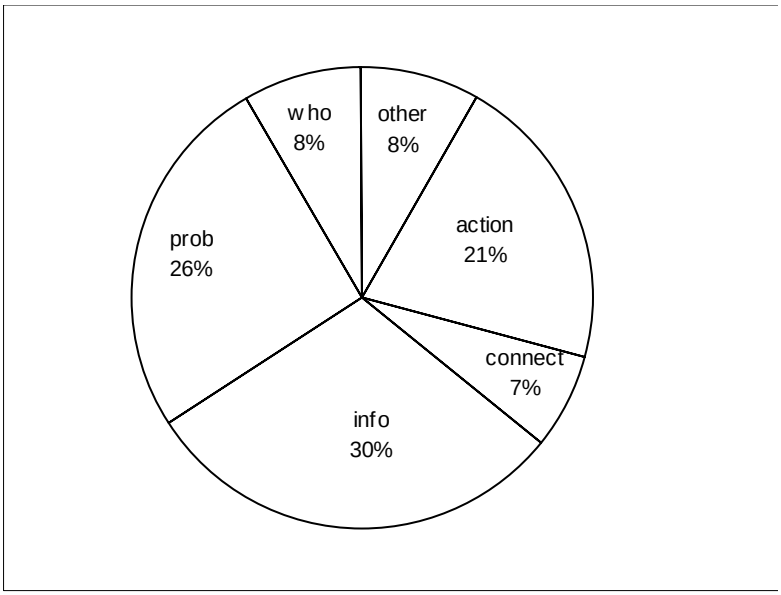

Figure 3.1 Distribution of primary move types in Oasis corpus

3.2.1.1 *Action* and *connect* move types

```
Oasis.1.tdef:phase2/day21/le769120,B,,,phase2/day21/le769120.1.msg,,action,0,~
@@ can you try er try a number for me @@ # it's permanently engaged i just
wonder if there's anyone on the line or whether it's a fault [ on the
line,ln,,,,,5,,,2,2.763,9.047,ok,good
```

Corpus Excerpt 3.2

Corpus Excerpt 3.2 shows another call transcription: this time the annotator has selected the *action* type. Although there is a problem statement, "it's permanently engaged", Table 3.3 makes it clear that the *action* type takes precedence. Furthermore, as BTexaCT (2001) points out, it would have been incorrect to interpret "Can you try…" as a question, of type *info*, as to what the operator is capable of doing. Rather, it is a polite way for the caller to request that the operator take a particular *action*, "try a number"; and it is the message that the caller wishes to convey, not a literal interpretation of the words chosen, that the annotation seeks to describe.

```
Oasis.4.tdef:phase5/day13/aa179940,B,,,phase5/day13/aa179940.1.msg,,info,0,yea
h hello uh this is not an emergency but ~ @@ can you connect me to the police
station please @@,dq,,,,,5,,,2,1.736,5.748,poor,good
```

Corpus Excerpt 3.3

Corpus Excerpt 3.3, however, shows that this policy was not always adhered to rigorously. In this clear-cut case of type *connect*, the annotator may have been influenced by the knowledge that the operator will answer that it is not possible; nonetheless, the caller's expectation is almost certainly that he will be connected, and other requests for connection to the police in the corpus are labelled *connect*.

The *connect* type is often used in situations where the caller explicitly asks to be transferred to another division of BT, such as directory enquiries or telegrams, or another telecomms provider. Occasionally – and rather marginally – it is used if the caller appears to have dialled the wrong BT number, presumably on the basis that the operator will offer to connect them, as in Corpus Excerpt 3.4.

```
Oasis.1.tdef:phase5/day05/aa128080,B,,,phase5/day05/aa128080.1.msg,,connect,0,
no ~ @@ it's er directory enquiries i want @@,dq,,,,,5,,,2,3.220,4.798,ok,good
```

Corpus Excerpt 3.4

If the caller asks to be connected to a telephone number, without specifying any problems they may have experienced dialling it themselves, the call is labelled *connect*. Where there have been difficulties, it is treated as a request for a line test, of type *action*:

```
Oasis.1.tdef:phase5/day04/aa124500,B,,,phase5/day04/aa124500.1.msg,,action,0,o
h hello um (hello) ~ i'm i'm having difficulty getting through to a number
wi... when i know perfectly well that the the phone is working but every time
i ring it i get a message telling me that the number's not been recognised
(alright sorry about that) @@ would you see if you could connect me
@@,ln,,,,,5,,,2,2.063,17.047,ok,good
```

Corpus Excerpt 3.5

3.2.1.2 *Who* and *info* move types

The *who* type is assigned when the caller asks to be given a BT number, or asks who they need to speak to. When they ask for a non-BT number, or for any other kind of information, *info* is used. Corpus Excerpt 3.6 and Corpus Excerpt 3.7 illustrate these distinctions.

```
Oasis.1.tdef:phase2/day01/le101980,B,,,phase2/day01/le101980.1.msg,,who,0,hell
o there i don't know if you can help me at all ~ @@ i'm trying to find out em
where abouts in {business}B {business}T i have to find out about er connection
to um the internet @@,internet,,,,,5,,,2,1.666,9.127,ok,good
```

Corpus Excerpt 3.6

```
Oasis.1.tdef:phase5/day01/aa106080,B,,,phase5/day01/aa106080.1.msg,,info,0,hel
lo i was just wondering ~ @@ could you tell me the number of um
{business}Waveney {business}Dog {business}Service dog wardens
@@,dq,,,,,5,,,2,2.099,7.015,ok,good
```

Corpus Excerpt 3.7

Corpus Excerpt 3.8 and Corpus Excerpt 3.9 show that occasional misclassifications do occur.

```
Oasis.1.tdef:phase6/day06/ab134350,B,,,phase6/day06/ab134350.1.msg,,who,0,~ @@
can you give me the number please for postal codes @@,other-
ref,,,,,5,,,2,1.838,5.938,poor,good
```

Corpus Excerpt 3.8

```
phase7/day55/ac141020,B,,,phase7/day55/ac141020.1.msg,,info,0,hi ~ @@ could i
er give me the number for international businesses
@@,idq,,,,,5,,,2,2.332,4.949,poor,good
```

Corpus Excerpt 3.9

The number "for postal codes" is provided by the Royal Mail, not BT. Corpus Excerpt 3.9 is a request for the (BT) international directory service. It is possible that these minor confusions arise because the *who* category is effectively a subset of the *info* category; as stated above, it is unlikely that they cause problems for the experiments reported in this thesis, because one would not expect the prosodic features of the two types to differ.

3.2.1.3 *Prob* and *other* types

Prob queries are generally straightforward, and annotation errors are rare. These calls appear to result from a caller having no clear idea what action they expect to be taken, and they are often long-winded. Corpus Excerpt 3.10 illustrates.

```
Oasis.8.tdef:phase7/day48/ac127590,B,,,phase7/day48/ac127590.1.msg,,prob,0,hel
lo there ~ @@ i've just had a bit of a funny experience on a with a phone call
(ooh) #! <laugh> @ not that good #! <laugh> @ i was just talking to my sister
and um @ i put the phone down and um @ the phone rung again (yeah) and um @
said hello obviously and @ said yeah what do you want and @ said i don't know
what do you want you rung me and @ said no i didn't you rung me so @ i said no
i haven't rung anybody and @ said well it's not me i haven't put any money in
so @ i said oh right you must have got the wrong number then @ so anyway and
put the phone down anyway @ i rung 1 4 7 1 and which probably should have been
me sister's last number who come up but @ it wasn't it was this other number
that come up so @ i don't know whether someone's well messing about or
@@,nuis,,,,,5,,,2,1.894,43.818,poor,good
```

Corpus Excerpt 3.10

Other queries generally consist of greetings (including cases where the caller's second, principal, utterance is also transcribed in the corpus), people calling the operator by mistake, crank calls, inexplicable calls and people claiming to be testing their telephones.

```
Oasis.9.tdef:phase8/day55/ad148120,B,,,phase8/day55/ad148120.1.msg,,,0,i've
got my green clothes on ~,other,,,,,5,,,2,1.695,3.437,poor,good
```

Corpus Excerpt 3.11

Observe that the move type field is left blank, not explicitly stated to be *other*.

This category contains about 20 incorrect annotations. Most of these appear to have arisen because the correct application-specific **call class** is indeed *other*, leading to confusion between the two classification systems. Examples are requests for the date and time, which should have been classified as *info*, and some lengthy statements describing telephone service problems, which were in any event not included in these experiments, because they were too long for the segmentation algorithm to handle.

With the exception of these very long utterances, the *other* calls are, typically, brief: this is because many consist of a greeting, an apology, or an obscenity, and nothing further.

3.2.2 Move-by-move composition of utterances

It has been stated that each utterance consists of at least a primary move, to which a type is assigned. More often than not, though, there is also a greeting or social move, as well as any number of secondary moves: these are not always explicitly marked out in the corpus, although they are sometimes (in 392 cases in the corpus) delimited by a single @. Thus, a typical social move is “Hello”, a possible secondary move “My phone’s not working”, with the primary move (because *action* takes precedence over *prob*) “Can you fix it please?” With this in view, how justifiable is it to treat the primary move type as a blanket descriptor of the entire utterance, as was done in this research?

An alternative approach might have been to excise the primary move from the utterance, and use only this part in the prosodic analysis. But, apart from the practical problems that it would have posed, the approach would have been flawed: some of the chunks would have been utterance-initial, others medial, others final, and one would expect to discover entirely different prosodic features in each position, so that the chunks would not be directly comparable. The approach would, for example, ignore (or be subject to artifacts deriving from) the downdrift phenomenon, discussed in Chapter 1.

Significant, too, is that BTexaCT (2001) instructs annotators to decide the type of the primary move with reference “to the whole utterance excluding the social move”. The primary move type is, therefore, intended to encode the force of the utterance *in toto*. Furthermore, in 5887 of the 8441 utterances in the corpus, the boundaries of the primary move correspond to those of the entire utterance, save the social move.

Since social moves generally consist of formulaic utterances, and vary little across utterance move type, it can safely be assumed that they will have no significant impact in prosodic analysis.

3.2.3 Corpus training and test sets

The sound files were supplied by the compilers in a directory hierarchy which grouped recordings made at the same time together. Because there could easily be artifacts which influenced a particular recording session, BT also divided the corpus into orthogonal segments, useful for forming test and training sets. Eight of these segments contained 1000 utterances apiece, with 441 in the ninth. 40 long utterances were removed from the corpus, as noted above.

The distribution of move types illustrated in Figure 3.1 is represented across the segments. In every segment, *info* is the most common category, followed always by *prob*, and *action* in third place. In all but one of the segments, *connect* is the least common class, with *other* and *who* generally ranking either fourth or fifth.

Experiments were conducted with a range of different training and test sets, as discussed in detail in Chapter 5. In general, though, three sets were used for initial training, three sets for secondary training (where this was carried out) and two sets for testing. The ninth, small, segment was set aside for final evaluation, after parameter tuning was complete.

3.3 Conclusion

This chapter has discussed the data options available for the research, justifying the decision to use the Oasis corpus. The annotation and transcription schemes used in that corpus are described, different call classes and primary moves are exemplified, and some minor compilation errors are set out. The composition of the corpus in terms of test and training sets, and the way such sets were used in the experimentation, is also described.

The next chapter gives an overview of the PLoNQ system architecture.

4. System architecture

Input data to the UT classifier, we have seen in the preceding chapter, is of two forms: an utterance, in the form of a sound file, and an annotation giving the UT of the utterance. The annotation is used in training the prosodic model and, in testing, to verify whether the classifier has done its job properly. It consists of a string from a text file and requires no pre-processing.

The sound file, on the other hand, requires considerable pre-processing, and it will be seen from Figure 4.1 that four distinct modules of the system are devoted to this task. The challenge here is to distil, from an extremely rich source of acoustic data, a meaningful and parsimonious set of prosodic features, which will be of use in distinguishing UTs. The first step, naturally enough, is to apply a **pitch detection algorithm** (PDA) to the speech wave in order to derive a pitch contour. The contour can be represented graphically; but what is needed for the UT classifier is a set of pitch readings at regular sampling points during the utterance. The choice of PDA and related issues are discussed in 4.1.

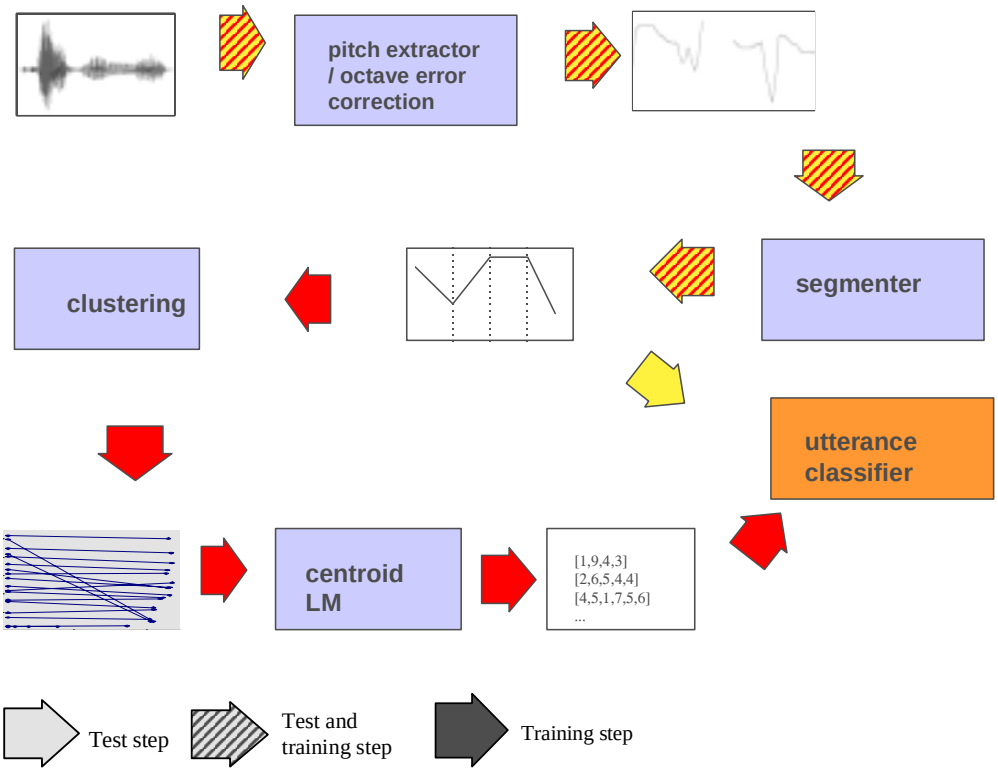

Figure 4.1 System architecture

Typically, a PDA produces an estimate of the pitch 60 times a second. Thus, one is rather swamped with raw data, and it is difficult to discern general patterns from it. What was needed was a way of observing the overall pitch topology of the utterance: a steep rise, perhaps, followed by a flat portion, then an unvoiced region, then a gentle fall. To respond to this requirement, a module called the **segmenter** (discussed in 4.2.1) was developed. The segmenter breaks the utterance up into a tractable number of prosodic segments – perhaps a dozen, for a medium-length utterance. The boundaries of each segment should correspond to a change in pitch gradient trend within the utterance: thus, for example, it should occur at a peak rather than halfway up a slope. The module provides for each segment a vector comprising three features: *duration*, *gradient* and *mean pitch*.

The next step in deriving a representation of these sets of features suitable for modelling is to find a small set of feature vectors which can act as an approximation of most of the segments found in the data: thus, all long gentle rises might be grouped together, as might all sharp falls. In other words, it was necessary to label the segments. For this purpose the **K-means clustering** algorithm, implemented by Lo Boon-hooi, was used. This module of PLoNQ is described in Section 4.4.1.

Thus, the sound file data is represented by an abstract set of labels indicating pitch pattern. Treated in isolation, though, these labels might not be particularly revealing, when it comes to the task of UT prediction. Just as in topic identification a trigram might be more indicative of a specific topic than a single keyword, so in this task one would expect better classification accuracy if sequences of labels, yielding information about longer stretches of speech, could be processed. Extracting such sequences is the task of the language model (LM) module, reported in Section 0.

The UT classifier module itself is described in Section 4.8.

4.1 Pitch detection algorithms (PDAs)

Pitch is difficult to measure objectively. It is obviously determined by a physiological phenomenon, namely the variation in tension, and thus the rate of vibration, of the vocal folds (Pickett, 1980:81), whilst the folds are **adducted**, or positioned for voicing. Opinions have differed on what is primarily responsible for the tension variation: Lieberman (1967) holds that it is caused largely by changes in subglottal air pressure (P_s), while Ohala (1978:13) believes that the interaction of two laryngeal cartilages, the thyroid and cricoid, is far more significant, and declares Lieberman's hypothesis “nonempirical”. Later studies, such as Laver (1994), indicate that both mechanisms contribute to vocal fold tension.

The rate of vocal fold vibration in speech is expressed in acoustics terms as **fundamental frequency** (F_0), which is defined by Ladefoged (1962) as "the frequency of repetition of a sound wave of which, when analysed into its component frequencies, the fundamental is the highest common factor of the component frequencies".

The term *PDA* is something of a misnomer, therefore. It provides an estimate of F_0, a strictly acoustic phenomenon. Pitch itself, though, is a matter of perception: it stands in the same relationship to F_0 as loudness does to RMS (root mean squared) energy. Unsurprisingly, the acoustic measurement does not always tally with what human listeners perceive, and what we see on an F_0 trace might not correspond to our intuitions on how an utterance was enunciated. F_0 estimation is a fairly error-prone procedure: it is fairly common for unvoiced regions to be marked as voiced by a PDA, and vice versa, for example. Another typical error is where a sample's F_0 is computed to be either precisely double, or precisely half, the true value. As Ying et al (1996) explain, although the F_0 has its origins in the rate of vocal fold vibration, the signal has to pass through the rest of the vocal tract. Resonances here may behave like band-pass filters, enhancing the spectral energy of harmonics at the expense of that of F_0. Thus, the PDA may deem the harmonics to represent F_0 estimates.

A number of pitch detection algorithms are described in the literature: the simplified inverse filter tracking algorithm (Markel 1976), the super-resolution (Medan et al 1991), auto-correlation (Hess 1983, 351-6), cepstral (Noll 1967), average magnitude difference function (Ross 1974). We experimented with the auto-correlation and cepstral approaches, as well as the integrated linear prediction/dynamic programming algorithm of Secrest & Doddington (1983), all of which were available under the Speech Filing System (SFS).

4.1.1 Auto-correlation

Some PDAs, including auto-correlation and AMDF (Average Magnitude Difference Function), involve extracting a fixed-length frame from the acoustic signal. Typically, according to Farinas (1999), the frame consists of either two or three periods, and a correlation is made between one

period and its neighbour. In auto-correlation, specifically, a comparison is made between a frame and a temporally shifting copy of itself. Krause (1984) explains how the boundaries of a fundamental frequency period are determined: a copy of the frame is compared to the original with a delay factor which is gradually incremented. When the delay factor is equal to 0 (that is, the original and the copy are aligned) the sums of products of each pair of samples are equal, and the auto-correlation value, which is the quotient of the sums, is therefore equal to 1. When the delay factor is incremented, the two frame representations are no longer aligned, and the quotient of the sums of products does not reach the maximal auto-correlation value, 1, until the next fundamental frequency period is encountered.

Farinas (1999) expresses the auto-correlation periodicity function as

$$FP_{AutoC}(\tau) = \frac{1}{n}\sum_{i=1}^{n-\tau} s_i s_{i+\tau} \quad (4.1)$$

In (4.1), $(s_i)_{i=1,n}$ represents the sequence of signal samples, and τ the delay factor. Farinas shows that the maxima of the periodicity function correspond to multiples of the fundamental period. The first peak (that is, the delay factor yielding the best correlation) gives the fundamental frequency.

4.1.2 Cepstrum analysis

In acoustic analysis, a spectrum is obtained by applying a Fast Fourier Transform to a speech wave form. When an IFT (Inverse Fourier Transform) is applied to the log of the spectrum, a **cepstrum** results, which for periodic or quasi-periodic wave forms, such as speech, gives a clear indication of fundamental frequency (Krause 1984). Time-domain algorithms such as auto-correlation tend to suffer from the halving and doubling mentioned above; for clean, relatively low-pitched speech, the frequency domain cepstral algorithm can be effective. In the case of the Oasis data, as we shall see, this was not the case; probably because of the telephone quality of the speech.

4.1.3 Integrated algorithm

Secrest & Doddington (1983) used a linear prediction approach, in which the next signal value is predicted from previous values, which are weighted with prediction coefficients. These coefficients aim to minimize the error between predicted and observed signal values (Krause 1984). Secrest & Doddington then employed dynamic programming as a post-processing technique, so that pitch candidates could be retained over several frames of speech before a final decision on F_0 was reached. Through the imposition of penalties, the algorithm seeks pitch candidates which differ minimally from preceding candidates, maintaining a sequence of back-pointers to achieve this.

4.1.4 Experimental comparison of PDAs

Figure 4.2 gives the SFS output for an utterance from the Oasis corpus, in six panes consisting of a speech waveform, a wideband spectrogram, the three F_0 contours, and a transcription which was added manually. The five-syllable utterance is enunciated in a slightly peremptory style, with a perceptible fall on each syllable except the last, which appears flat. There are no rising contours, which might suggest enquiry or hesitation: the caller is quite sure of himself and wants some action taken (although the nature of the action will probably not be clear to the operator, in this case). The overall pitch pattern described is represented satisfactorily by all the PDA algorithms, for the most part.

Whenever we see a sudden and brief rise or fall in the F_0 contour we can assume that the contour is wrong – especially if the excursion is to a point approximately half or double the frequency of preceding or successive samples, evidence of an octave error. Changes of pitch in speech production are, after all, simply not that abrupt. There may well be such an error at 0.48 seconds on the cepstral curve; on the auto-correlation curve, there are several halving errors in the first two syllables, and an instance of doubling at 0.91 seconds.

In Figure 4.2, the regions that one would expect to be voiced are grey-shaded in the pitch curve panes. It is clear that the integrated algorithm has performed best here, except for the final fricative, where the auto-correlation curve is unreliable, even though it does indicate voicing. The voicing indicated by the integrated algorithm at the beginning of the third and fourth syllables, outside the shaded area, is probably due to early voice onset time.

Other researchers, such as Taylor (2000) have found the Secrest & Doddington algorithm to be more accurate than the alternatives, especially for pitch tracking of telephone speech. Since this comparative experiment led to the same conclusion, it was decided to adopt the integrated PDA in this research.

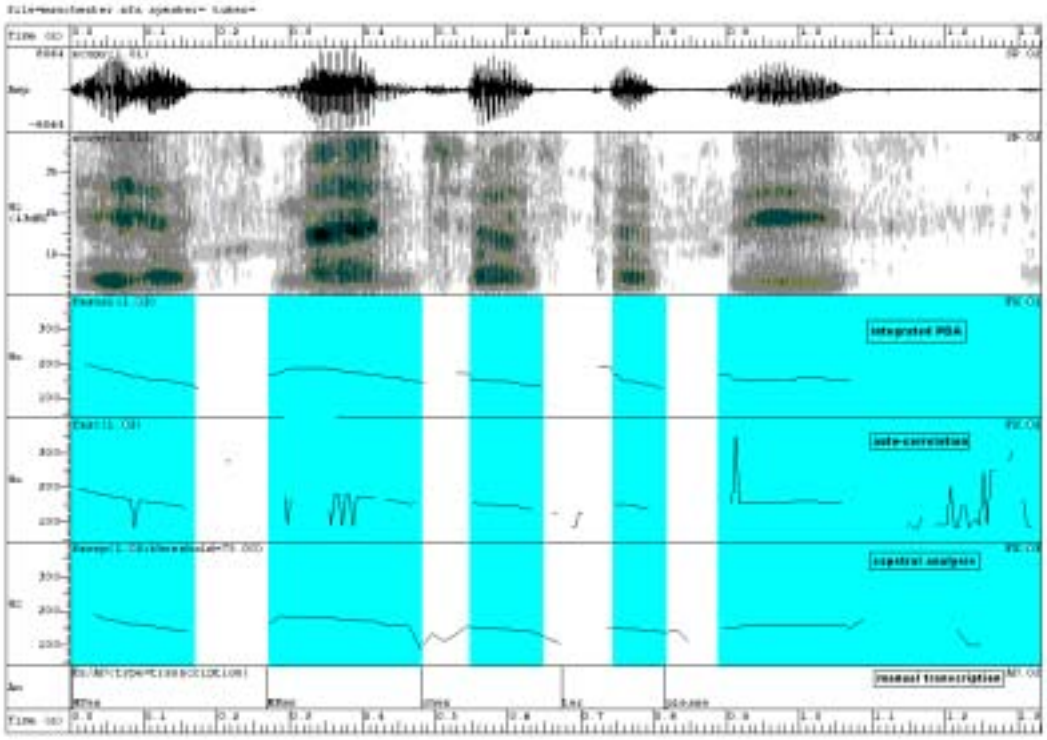

Figure 4.2 Spectrographic and F_0 traces (using three different PDAs) from SFS, for utterance "Yes, Manchester please."

4.2 Pitch stylization

At this point, each corpus utterance is represented by a set of sequential F_0 samples: often, several hundred per utterance. The task now was to conflate contiguous sample regions to form segments of reasonable length. What constitutes "reasonable length" is difficult to determine objectively: clearly a ratio of one segment per sample would not be suitable, nor would one segment per utterance. What is needed is something between these two extremes, a stylization which can capture the overall rise-fall tendencies of the utterance, ignoring micro-prosodic effects, such as prosodic characteristics of the phonemes used by the speaker. These, according to Hirst et al (2000), are associated particularly with plosives and fricatives; Pijper (1983:17) writes that such effects may also be due to "imperfections in the articulatory mechanism".

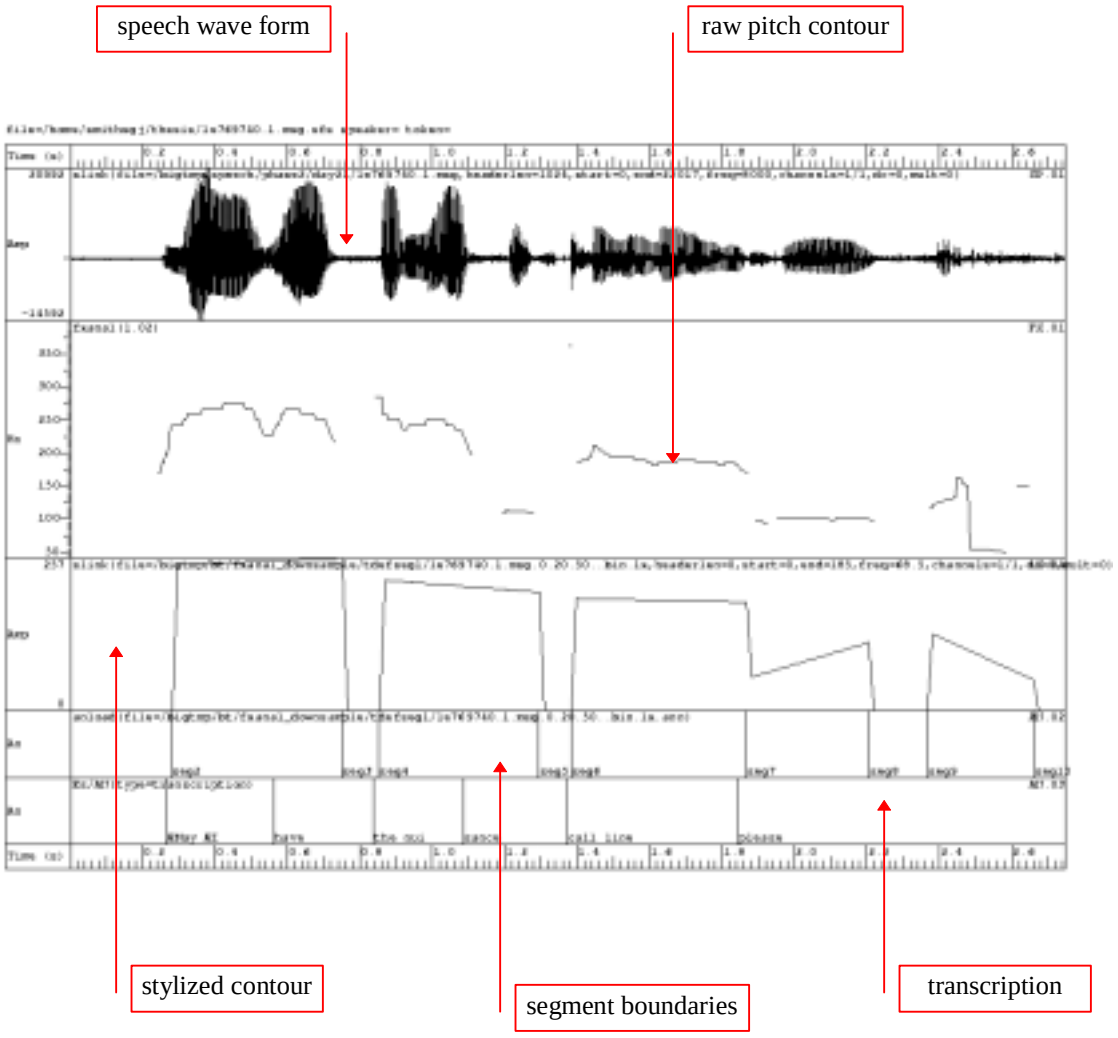

Figure 4.3 Segmentation of "May I have the nuisance call line please?"

Figure 4.3 includes the segmentation of an utterance from the Oasis corpus. Proceeding from top to bottom, the first pane of the figure shows the speech wave form of the utterance. Below this is the pitch contour output by the PDA, gaps in the black line indicating an unvoiced region. Next comes the segmentation, which, it will be seen, serves to stylize the pitch contour with a sequence of straight lines. The pane below shows the location of the segment boundaries, for cases where this is not clear from the stylized contour. Lastly, a transcription of the utterance has been added manually.

Observe from the raw pitch contour that over the course of the utterance, there is an overall pattern of declination, with a slight rise at the end, as might be expected in a yes/no question. The topology of the utterance is quite well represented in the stylized contour. In the seventh segment, the stylized contour rises slightly, while the raw contour appears to be flat; this is because the two very short unvoiced regions in the vicinity have been incorporated into the segment itself, reducing the average pitch at the beginning.

In this example, segment lengths were restricted to a minimum of 20 samples (≈0.3s) and a maximum of 50 samples (≈0.7s), except in unvoiced regions, so that there more segments than words. Experiments in which segment length was constrained to different extents are reported in Chapter 5.

4.2.1 Segmentation algorithm

The algorithm used to perform the segmentation makes use of linear regression and dynamic programming. An exhaustive search of every available segmentation scheme is first made – so, in an imagined utterance consisting of F_0 samples numbered 1, 2 and 3, the following segmentation schemes would be considered:

a) [1 | 2 | 3]
b) [1 2 | 3]
c) [1 | 2 3]
d) [1 2 3]

In a real utterance, with many pitch samples, the number of segmentation schemes of course grows exponentially; this is why some 40 very long utterances from the corpus were not treated in this thesis. In these cases, the segmentation algorithm would have taken some hours to process just one utterance.

For each segmentation scheme postulated, the segments within it are considered. Within each segment, the line of best fit between the data points (the F_0 sample readings) is computed by linear regression, as shown in Figure 4.4. The average distance of all data points from the corresponding position on the line of best fit is then calculated.

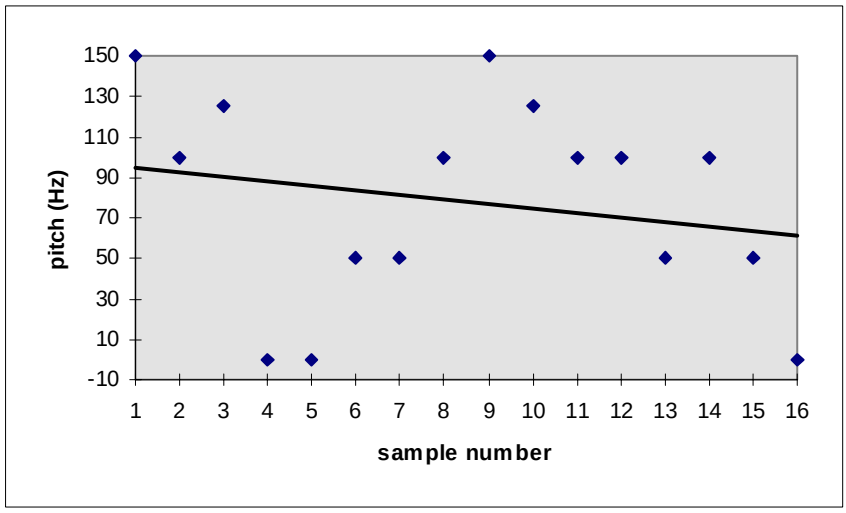

Figure 4.4 Data points for one (imaginary) segment showing the line of best fit

Under one possible segmentation scheme, each segment consists of just an adjacent pair of samples. Because in this case both data points lie on the regression line, no error will arise, and the scheme will always be selected by the program as optimal. There are two features for ensuring that segments are of reasonable length. The minimum segment length was referred to above; and there is also a **segment insertion penalty**, which taxes postulated segments in inverse proportion to their length. Values for these features are specified as arguments when the segmenter is run, and experiments with different values are reported in Chapter 5.

Under the segmentation algorithm, a candidate segment with start time *s* and end time *t* is assigned a segment instantiation cost D(*s*,*t*). If the segment violates the minimum and maximum length constraints, D(*s*,*t*) is set to an arbitrarily large number; otherwise it is defined as (4.2).

$$D(s,t) = \sum_{r=f0}^{t} (f0(r) - \hat{f}0(r))^2 \qquad (4.2)$$

In (4.2), *r* represents intermediate sample times in the candidate segment, and $f \leftarrow 0(r)$ is the distance of the F_0 sample from the segment's linear trajectory at time *r*.

A cumulative candidate segmentation scheme is computed as per (4.3), where $C(t)$ is the accumulated cost of the scheme up to the end of the current segment. *K* represents the segment insertion penalty; it will be seen that the total cost of a scheme will increase with the number of segments it posits. Because short segments are naturally numerous, longer segments are thereby rewarded.

(4.3) $$C(t) = \min_s \{ D(s,t) + C(s-1) + K \}$$

Once $C(t)$ has been accumulated for the whole utterance (that is, time *t* is utterance-final), scheme costs are computed for all the other potential segmentation schemes. That with the lowest cost is selected.

4.2.2 Linear regression in stylization: other work

Scheffers (1988) also made use of linear regression to build stylized pitch contours. His goal was to suppress excursions in the F_0 contour which did not affect perceived pitch. He resynthesized utterances from his output contours, playing the results to subjects in evaluative experiments. He found that in many cases the subjects could not distinguish utterances synthesized from two sources: the original F_0 contour, and the stylization.

Scheffers's model attempts to find **turning points** (segment boundaries) in the F_0 contour which represent significant changes of pitch. Each F_0 sample is examined, and if it diverges by more than *maxvar* from the pitch predicted by the extension of a linear trajectory from previous samples, it is a candidate turning point; *maxvar* is the maximum expected excursion of F_0 due to micro-prosodic events. Another variable, *mindur*, governs the "minimum duration of intonationally important movements", and determines the number of samples exceeding *maxvar* before a turning point may occur.

Bagshaw (1994:164) implemented Scheffers's algorithm, but computed the least median (as opposed to mean) of squares in the linear regression, maintaining that this approach is less sensitive to outlier data points. Bagshaw also aligned turning points with syllable nuclei (he used an automatic syllabification tool as a pre-processing step), to ensure that no turning point corresponded, because of PDA error, to an unvoiced region of the utterance.

An alternative stylization technique, **MOMEL**, was developed by Hirst et al (2000). Experiments using this technique are reported in Chapter 7, and the results compared with those of the equivalent modules of PLoNQ.

4.3 Normalization of stylized contour

The Oasis corpus contains calls from people of all ages and both sexes. It is to be expected, therefore, that the pitch dynamic range of callers will vary tremendously. Now, it was noted above that the output from pitch stylization is a vector of three features – mean pitch, slope and duration – which is later input to a clustering algorithm to generate a set of prosodic labels. Of the three features, mean pitch is undoubtedly the most susceptible to inter-speaker variation: in making use of absolute pitch, one could be accused of modelling characteristics of the speakers, rather than of the utterances. This could, of course, apply in the case of the other two features, since some people speak more slowly, or in a more monotonous fashion, than others.

When the feature vectors of the prosodic labels were examined, however, it was found that a number of the labels differed from each other largely with respect to mean pitch, the other two features having very similar values. This demonstrated that the primary inter-speaker distinguishing feature was indeed pitch; and there was a risk that labels might be assigned to unseen segments largely on the basis of their absolute pitch, rather than genuine prosodic data. It therefore seemed appropriate to normalize all the pitch features before passing them to the clustering algorithm. This was done by simply averaging the mean pitch values in each utterance, and subtracting that average from each value.

Bagshaw (1994) used a different normalization scheme in his experiments. He plotted the line of

best fit through all turning points in the utterance, subtracted the modelled value from the actual F_0, and divided the result by the standard deviation.

Whichever scheme is used, it must be borne in mind that in a corpus such as Oasis, where there is generally only one utterance per speaker, the evidence on a speaker's dynamic range is probably inadequate. Some UTs may vary intrinsically in average pitch, so that if there were a known instance of some individual calling twice – first with a request for action, say, then later to report a problem – the average pitches could vary substantially.

PLoNQ does not attempt explicit modelling of perceived pitch. In such regions, the PDA reports an F_0 value of 0Hz, and at the stylization stage these regions go largely unchanged, because of the absence of length constraints on them in the segmentation algorithm: absence of voicing is thereby effectively modelled. With normalization, however, the speaker's average pitch is deemed to be 0Hz, and absence of voicing is marked by a negative number that varies from speaker to speaker. In this research, experiments have been conducted both with and without normalization, and the results compared.

4.4 Clustering

Clustering is a mechanism for finding natural sets (clusters) for equivalent classes in data. In this work, it is used to assign set membership, by grouping data with similar parameter values. Where each data item is associated with only one value, one could simply partition the data into groups; but with a multi-dimensional feature vector, such as our prosodic segment descriptions, the power of clustering can be turned to full advantage.

Because of the **curse of dimensionality**, in fact, it is not really practical to extend the partitioning approach to more than one dimension. When working in one dimension, it is often enough to divide the axis into equally long intervals; but in the multidimensional case, the number of quantization points goes up exponentially. In natural data, moreover, the points are unlikely to be uniformly distributed, so this simple approach is inefficient.

Clustering algorithms have been used in fields as diverse as clinical diagnosis (Beauchaine & Beauchaine 2002), climatology (Gong & Richman, 1995) and data compression (Zhang et al 1997), as well as speech processing. Bacchiani (2000) used maximum likelihood linear regression clustering for speaker recognition, and Nakai et al (1996) clustered F_0 contour parameters to detect phrase boundaries in Japanese.

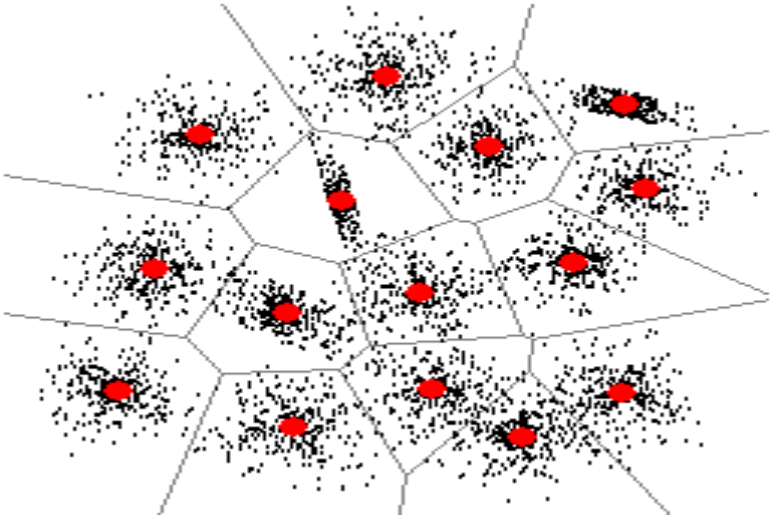

Figure 4.5 Clustered data points in two-dimensional space: an impressionistic view

Figure 4.5 gives an impressionistic view of arbitrary data points in two-dimensional space, along with a likely cluster membership configuration.

The input to a clustering algorithm is the raw data; output consists of the number of data points assigned to each cluster, as well as the cluster's **centroid**, which is the mean of all parameter values in that cluster.

Two broad classes of clustering algorithm, described by Hartigan(1975), are **hierarchical** and **non-hierarchical**. An example of the first class is the **minimal spanning tree clustering algorithm.** This algorithm initially assumes that each data point is a cluster in its own right. The cluster which is closest (in Euclidean terms) to some other cluster is located, and these two clusters are merged. Iterations of the algorithm take place until sufficient merging of clusters has occurred for the user-specified number of clusters to be reached. Distance between two clusters A and B may be computed from either their centroid's positions, or the positions of the members of A and B which minimize the distance. As well as cluster membership assignment, the algorithm can provide a tree diagram of membership evolution. Non-hierarchical algorithms, on the other hand, make available the user-specified number of clusters from the outset. Hartigan showed that non-hierarchical approaches are in many cases more efficient, as they require fewer algorithm iterations before convergence is reached.

4.4.1 *K*-means clustering

The (non-hierarchical) *K*-means clustering algorithm described by Bengio (1996) was used for our experiments. The algorithm proceeds as follows:

i) K clusters are initialized with a prototype vector chosen more or less at random from training data

ii) For each training data feature vector, the nearest prototype vector is located, and provisional cluster membership established

iii) The centroid position (that is, the mean of all training data feature vectors) is computed for each provisional cluster, and it takes the place of the prototype vector in step ii.

Steps ii and iii are repeated until convergence occurs; that is, the centroid vectors computed in two iterations of step iii remain the same. The mean of the *k*th cluster s_k is computed by (4.4), where μ_k^n and x^n are the *n*th component of the mean μ_k and feature vector *x* respectively, and N_k is the number of elements in the *k*th cluster. In our case the component (*n*) corresponds to segment slope, duration and average F_0.

$$\mu_k^n = \frac{1}{N_k} \sum_{x \in s_k} x^n \tag{4.4}$$

The sum of least squares clustering function used to calculate error *J*, the distortion between the vector quantized centroids and the data. In (4.5), K is the number of vector quantized centroids.

$$J = \sum_{k=1}^{K} \sum_{x \in s_k} \left\| x - \mu_j \right\|^2 \tag{4.5}$$

where

$$\left\| x - \mu_j \right\|^2 = \sum_{n=1}^{D} (x^n - \mu_j^n)^2 \tag{4.6}$$

and *D* is the dimensionality of the feature vectors, in our case 3. The outcome of this process is a set $\{\mu_1, \ldots, \mu_k\}$of cluster centres which minimize distortion *J*. These constitute the basic prosodic labels.

4.5 Secondary training

In this component of the system, a second set of training data is treated as unseen material, and prosodic labels assigned according to the evidence of the initial training set. Pitch extraction, segmentation and optional normalization are applied to this data; then, for each three-dimensional feature vector in the data, the nearest cluster centroid $\hat{c}_k$ (from the set of centroids in the **primary** training data) is computed by (4.7). The mean and variance of the *k*th centroid, in the primary training data, are denoted by m_k and σ_k respectively, while x_k is the observed feature vector from secondary training data.

$$\begin{aligned} \hat{c}_k &= \arg\max_k P(x_i|c_k) \\ &= \arg\max_k \sum_{n=1}^{N} -\frac{1}{2}\log {\sigma_k}^n - \frac{(m_k^n - {x_i}^n)^2}{2{\sigma_k}^n} \end{aligned} \tag{4.7}$$

The two-tier training data approach helps to preserve data independence. If, having derived a set of labels from data, one simply applies those labels to the same data and uses the result to classify unseen utterances, there is a risk that the classification will be skewed towards features of the training set – especially if that training set is not large, when there is a risk of overfitting. Experiments were, however, performed both with and without the intermediate training stage, and the results did not differ as widely as might have been expected, as reported in Chapter 5.

In the experiments without secondary training, labels (centroids) are assigned to the segments (feature vectors) of the **primary** training data instead (computed from cluster means and variances also of the primary training data).

Once the primary or secondary training data has been labelled by means of (4.7), the labels are passed forward to the prosodic label sequence modelling module, described in the next section.

4.6 Prosodic label sequence modelling

Fox (1984) advocates an approach to prosodic analysis which examines sequences of tone-groups (segments) in the quest for links between intonation and discourse structure; He holds that "the tone-group is not an isolated entity, but contracts relationships with other tone-groups in sequences". It is, in natural language processing generally, quite usual to make use of **n-gram** based language models. In a topic identification experiment, for example, the predictive power of the **bigram** "Fleet Street" would be much more compelling than that of the simple **unigrams** "fleet" and "street", weighting the classifier in favour of a news or media category.

In our classifier, label sequence modelling is implemented in a straightforward way. The user may specify what n-gram order is required; if a short utterance consisted of prosodic labels 2 - 4 - 6 - 8, and the user requested an n-gram order of 1 to 3 (1:3), the following sequences would be processed:

a) 2
b) 2 4
c) 2 4 6
d) 4
e) 4 6
f) 4 6 8
g) 6
h) 6 8
i) 8

It can sometimes happen that, where the specified n-gram order is high, sequences present in the test (or secondary training) data may be extremely rare in the initial training data, making it difficult to estimate the probability of occurrence. Moreover, if the sequence is not found at all, a probability of zero would by default be assigned. This has serious computational consequences: since class membership of an unseen utterance is determined from the product of class probabilities in training data, there is a risk that some class could be completely ruled out, simply

because a particular long sequence is not represented in that class's training data.

In some other work, such as the dialogue classification of Wright (2000), a technique known as **backing off** is used with the n-gram LM. Under this technique, the probability of the n-1 gram, multiplied by a weighting factor, is used to estimate the probability of a rare n-gram. Thus, if the probability of the sequence 3 - 5 - 7 turns out to be zero, or is below some user-specified threshold, then the weighted probability of the sequence 3-5 is used instead. The effect of the weighting factor is to re-distribute some of the probability mass from high frequency n-grams to those of low frequency, or which do not occur in training data.

PLoNQ does not make use of backing off, for two reasons. In the first place, the problem of zero probability for an observed sequence belonging to a particular class is dealt with setting such a probability to an arbitrarily small value; experiments aimed at determining the optimum **floor value** are described in Chapter 5.

Secondly, in PLoNQ, the user specifies an n-gram order range in the format *min*:*max* (for example 1:3 requests treatment of unigrams, bigrams and trigrams, while 2:2 looks at only bigrams). Notwithstanding this flexibility, one would not expect that leaving lower-order n-grams out of account would of itself help classification accuracy, so the experimentation was conducted with a *min* value of 1. Thus, the influence of lower-order n-grams is taken into account in all cases anyway, and there therefore seemed little point re-apportioning probability mass to such n-grams.

4.6.1 Acoustic morpheme analysis

The n-gram modelling of prosodic label sequences was inspired by the research of Gorin et al (1999) on **acoustic morphemes**. In earlier work (in particular Gorin 1995), they had described **HMIHY** ("How may I help you?"), a system not too dissimilar from our own, which can automatically transfer people calling the AT&T operator to the correct department. The earlier implementation performed on-line transcription by means of a speech recognition module, then used a salience algorithm to detect keywords, or groups thereof, such as "credit card call" or "area code", routing the call on this basis.

The later implementation was somewhat more ambitious in that it relied on salient sequences of phones – acoustic morphemes – to perform the classification. The advantage of this approach, of course, was that it avoided the need for transcription of the data: if the classes can be computed directly from phonetic information, it is possible to avoid the word error rate problem associated with the speech recognition task.

In Gorin et al's work, sequences of four or fewer phones, with an absolute frequency in the training set of 5 or more, were selected, and further filtered for salience and mutual information of their components. When applied to the classification task, the phone sequences discriminated between call types almost as well as earlier experiments involving keywords, although the false rejection rate (where the system was unable to ascertain that the call belonged to any of the classes) was markedly higher.

Since phones are constituents of words, this result was perhaps not entirely unexpected. We decided to adopt Gorin's approach, exploiting sequences of prosodic labels instead of acoustic morphemes, in the belief that while an isolated label, like a phone, would serve no discriminatory function, a sequence of such labels might capture more of the overall prosodic pattern of the utterance and, potentially, its meaning.

4.7 Data pruning

The next sections describe two data pruning, or thresholding, techniques that were applied to the PLoNQ classification. The techniques have been presented in Chapter 2, and, as the reader will recall, they attempt to ensure that only significant training data which discriminates between classes is taken into account by the classifier. Training data without this relatively strong predictive power is disregarded.

By a further data pruning technique, only label sequences that occur more than a specified number of times in the training data are used. If a sequence occurred only once, for example, its predictive power would be weak because its class membership would not reflect any real statistical pattern. Experiments with all three techniques are presented in the next chapter.

4.7.1 Mutual information (MI)

Oakes (1998:63) reports that co-occurrence statistics such as mutual information "are slowly taking a central position in corpus linguistics". MI provides a measure of the degree of association of a given segment with others. Other research (such as Kobayasi et al (1994) for Japanese, and Lauer (1995) for English) has demonstrated the value of MI in detecting compound status, by determining the strength of association between collocates, while certain corpora, such as the Academia Sinica tagged corpus of written Chinese (Academia Sinica 1998, Smith 1999), come pre-annotated with MI scores for collocates. Gorin et al (1999) give (4.8) as an approximation of an acoustic morpheme's pointwise mutual information – "a measure of its utility for recognition". In his experiments, acoustic morphemes (*f*) with MI less than 1 are filtered out.

(4.8) $$I(f) = I(p_1 p_2 \cdots p_{n-1}; p_n)$$

According to (4.8), all sequences, whether of two constituents or more, are divided into two parts: the final constituent, and all the other constituents. Pointwise MI, calculated by (4.9), is what is used in lexical processing to return the degree of association of two items (a **collocation)**.

(4.9) $$I(x; y) = \log \frac{P(x|y)}{P(x)}$$

Where one constituent of a collocation could scarcely occur other than in the company of the other (as with "Hong Kong", perhaps), MI will be positive and relatively high. Zero MI indicates, in principle, that two items are contiguous by chance, and that they are independent of each other (although it is quite difficult to make out a case for independence when word order is clearly constrained by rules of syntax). A negative MI suggests that the items are relatively common, but in complementary distribution: ungrammatical sequences such as "the and" would come into this category.

By extension, mutual information can usefully be applied too to sequences of phones or prosodic labels. In the classification, we would wish to process items which exhibit a pattern of co-occurrence with certain other items, paying less attention to those pairs associated only by chance.

4.7.2 Salience

We also followed Gorin in adopting the maximum of the a posteriori distribution (4.10) as a measure of the salience of a sequence for the classification task. *C* is the UT class.

(4.10) $P_{max}(f) = \max_{C} P(C|f)$

Salience as a data pruning technique was discussed in Chapter 2. Experiments with various salience thresholds are described in the next chapter; Gorin selected sequences with a salience greater than 0.5 in his research.

4.8 The classifier module

The classifier module calculates the probability of each sequence of prosodic labels f_n in a given utterance *u*, as shown in **(4.11)**.

(4.11) $P(C_u) = \arg\max_{c} \prod_{n=1}^{N} P(C|f_n)$

Where the sequence does not occur in training data for some class, the class conditional probability $P(f|C)$ defaults to the floor value mentioned in 0. The posterior probability is then derived from Bayes' law, as shown at (4.12).

(4.12) $P(C|f) = \frac{P(f|C)P(C)}{P(f)}$

P(*f*) is the number of tokens of a particular sequence in the whole of the training data (in principle divided by the total number of tokens of any sequence, but this is a constant); $P(f|C)$ is the number of tokens of the given sequence in the class divided by the total sequence count for the class, and $P(C)$ is the class prior, the number of utterances in the class divided by the total number of utterances.

4.9 Summary

This chapter has described the modules of the classification suite, tracing a path through the system architecture depicted in Figure 4.1. Some of the variables and parameters associated with the different system components have been mentioned in passing; the next chapter reports on optimal settings for them, and presents experimental results.

5. Experimentation and results

Much of the experimentation presented in this thesis consists of adjustments to parameters which affect the operation of programs which together make up the classification tool suite. As detailed in the previous chapter, these programs are piped together (the output from one serves as input to the next) in modular fashion. Parameter adjustments made at an early or intermediate stage of the process may be expected to have just as significant an impact on final results as those made with respect to the final classifier module. What is expected of the system is that it should maximize the number of correct classifications of unseen utterances, and it should be tuned in such a way as to reflect this.

One need only consider a system comprising ten modules, each of which avails of a mere ten parameter settings, to conclude that an exhaustive search for the optimal classification profile would be computationally intractable: one would need to conduct 10^{10} separate experiments to arrive at a decision. What has been adopted instead is a "best first and forward" approach, where a range of reasonable settings for each module are justified on intuitive or empirical grounds, before the results are piped forward. The approach is not, however deterministic to the point that one setting is deemed optimal and irrevocable with respect to each module. Optimal values for clustering parameters, for example, are ultimately determined empirically, by experimentation.

Given appropriate refinement of algorithm implementations, a more exhaustive set of experiments might have been conducted. In this way, the system parameters could have been determined in an entirely empirical manner, without recourse to linguistic reflection. It is, however, appropriate that the tuning of the system should be motivated by inspection and theoretical validation of intermediate results; if such tuning were conducted solely on the basis of final classification results, treating the system as a black box, then the research contribution of this work would surely be diminished.

If the goal of the work had been to maximize classification performance for a commercial application, perhaps, then theoretical and intuitive considerations would have been less relevant. It might have been possible to improve performance by an exhaustive experimental approach, but in the process we would have learnt less about the utility of the early and intermediate modules for prosodic analysis, and consequently the resulting tuned system would be less likely to generalize to other classification tasks.

In PLoNQ, two categories of system parameters are defined. The early and intermediate modules are known as **training parameters**; their distinctive feature is that the impact of adjusting them can be assessed independently of the final classification performance obtained. For example, it is meaningful to inspect a stylized pitch contour and ask ourselves how well it appears to mirror the raw output from the PDA (pitch detection algorithm). The tuning of the so-called **test parameters**, on the other hand, relies mainly on the comparative performance of the classifier module at different settings, and is considerably less amenable to a linguistic analysis. For example, the parameter *wprior*, the class prior weight, is set to a number between 0 and 1 that maximizes performance; there is no intermediate output for us to inspect which might lead us to prefer a setting of, say, 0.85 over 0.875.

Table 5.1 and Table 5.2 summarize the parameter setting choices available to the system, as discussed in Chapter 4; the optimal setting used to obtain the maximally correct final results is also given in each case.

Table 5.1 System modules and training parameter settings

module	parameter	available settings	optimal setting
pitch determination	algorithm used	cepstral auto-correlation integrated	integrated
pitch determination	pitch correction	binary	on
segmenter	segment insertion penalty	variable	90000
segmenter	maximum segment length	variable	∞
segmenter	minimum segment length	variable	20 samples
clustering	number of centroids	variable	30
clustering	normalization	binary	off
clustering	secondary training	binary	on

Table 5.2 System modules and test parameter settings

module	parameter	available settings	optimal setting
n-gram model	n-gram order	variable	1:3
classifier	minimum n-gram occurrence	variable	2
classifier	class prior weighting	variable	0.875
classifier	floor value	variable	10^{-12}
classifier	mutual information	variable	$-\infty$
classifier	salience	variable	0

5.1 Training parameters

5.1.1 Segmenter optimization

Labelling schemes which have not been fully automated, such as ToBI and Tilt (described in Chapter 1) make it their goal to describe specific sub-utterance units, whether at syllable level, for example, or motivated by a syntactic analysis, assigning a pitch shape to major constituents of the utterance, such as clauses. The PLoNQ segmenter cannot identify such sub-utterance units, of course; but given appropriate constraints on duration, a comparable effect may be seen in certain segmentations, such as those postulated for the utterance “Yes, Manchester please” shown in the following two figures.

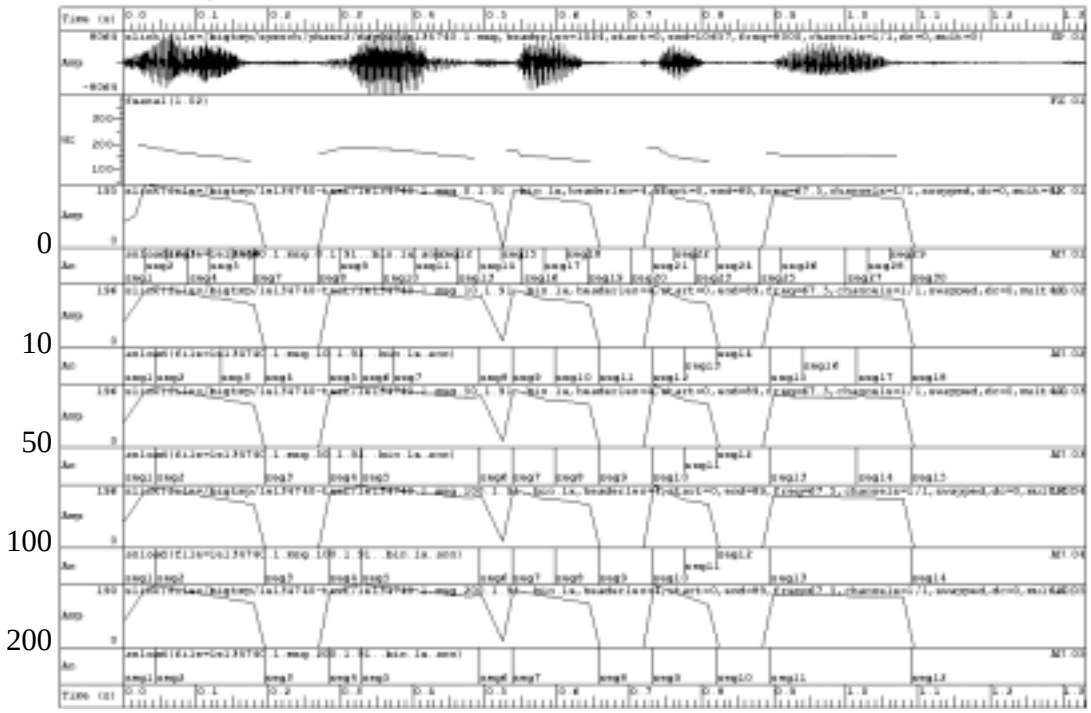

Figure 5.1 SFS output for "Yes, Manchester please" The first two panes show the speech waveform and PDA output; the other contours shown represent stylizations with the segment insertion penalty shown in large type on the left. Under each stylization, its segment boundaries are marked out.

Figure 5.1 shows possible segmentations of the utterance without constraints on segment length, and with various relatively low segment insertion penalties (as discussed in section 4.2.1). The third pane of Figure 5.1, after the speech wave and PDA output, shows the segmentation proposed with segment insertion penalty 0. This should match the PDA contour exactly, as the segmenter tries to find the fit to that contour with minimal error. Where the two differ, the PDA has calculated values for F_0 for an isolated sample which is not shown in its graphic output, but which is taken into account in the segmenter output: this phenomenon can be observed right at the beginning of the utterance, and just after 0.5 seconds. Another apparent discrepancy arises because the segmenter outputs a contour even for segments which are entirely unvoiced, and at boundaries shows a steep cline from the end of one segment to the first sample of the next. This accounts for the distinctive trapezoid shape of the contour in voiced regions. In most cases, however, the segments are sub-syllabic and too short to be useful.

5.1.1.1 Segment insertion penalty adjustment

Recall from the previous chapter that the segment insertion penalty is designed as a tax on short segments. As the segment insertion penalty is incremented, it will be observed, the number of segments decreases. Even with the penalty set to 200, though, the overall contour shape is preserved.

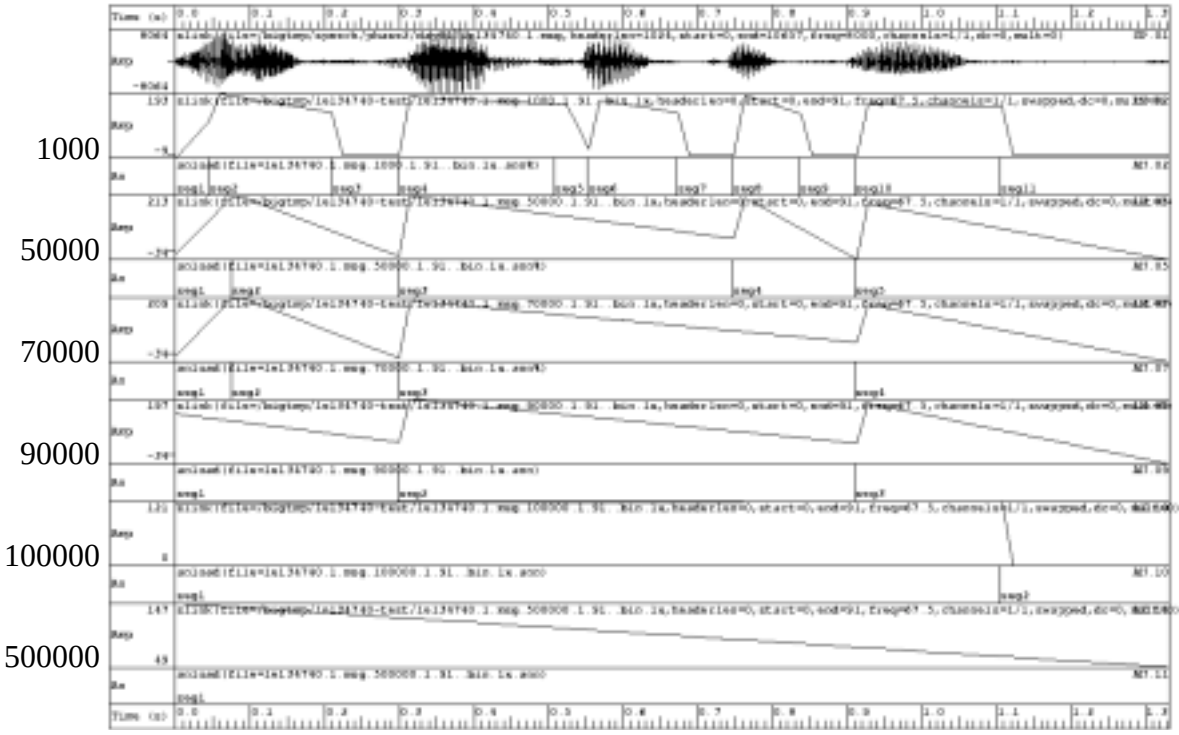

Figure 5.2 "Yes, Manchester please" with segment insertion penalties as shown on the left

Figure 5.2 shows output from the same utterance file with higher values for the segment insertion penalty. Where this is set to 1000, as in the second pane, there is a voiced segment for each syllable of the utterance, with further segments representing unvoiced regions. At 50000, the unvoiced and unvoiced regions are conflated into larger segments covering approximately one syllable, while at 90000 the segments map to one word. This seems to be the optimal segmentation for this particular utterance: the output with a segment insertion penalty of 100000 seems quite uninformative, and the single segment at 500000 merely tells us that there is downdrift over the course of the utterance, which, it was noted in Chapter 1, is generally true of all utterances.

It is not being suggested that the segmenter is generally capable of marking out word units, nor even that a segmentation based on word units would necessarily always be optimal. The ideal segmentation scheme for an utterance will, however, yield segments that begin and end at linguistically meaningful locations, such as word or clause boundaries. Figure 5.3 presents further evidence of this. The word “oh” makes no semantic contribution, so assigning it a segment of its own seems unjustified; the 90000 segmentation is therefore preferable to the 80000. With a

segment insertion penalty of 100000, the words “call” and “please” are assigned, rather implausibly, to a common segment.

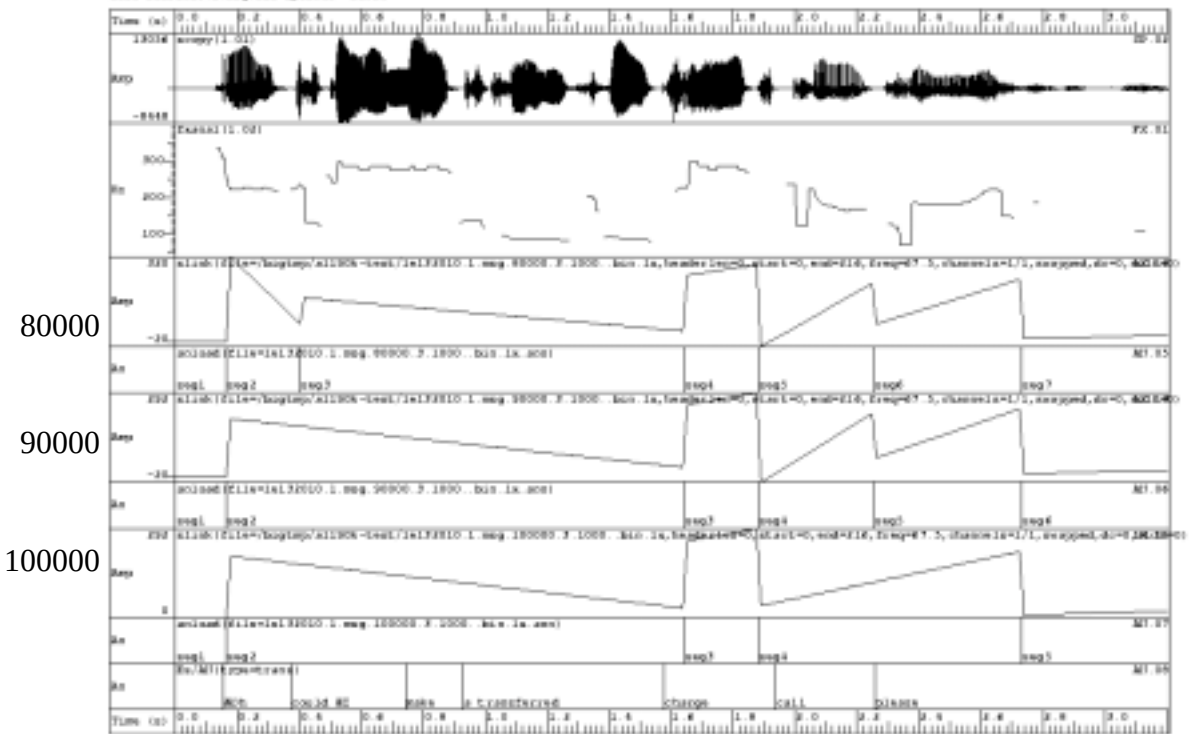

Figure 5.3 "Oh could I make a transferred charge call please?", with segment insertion penalties as shown

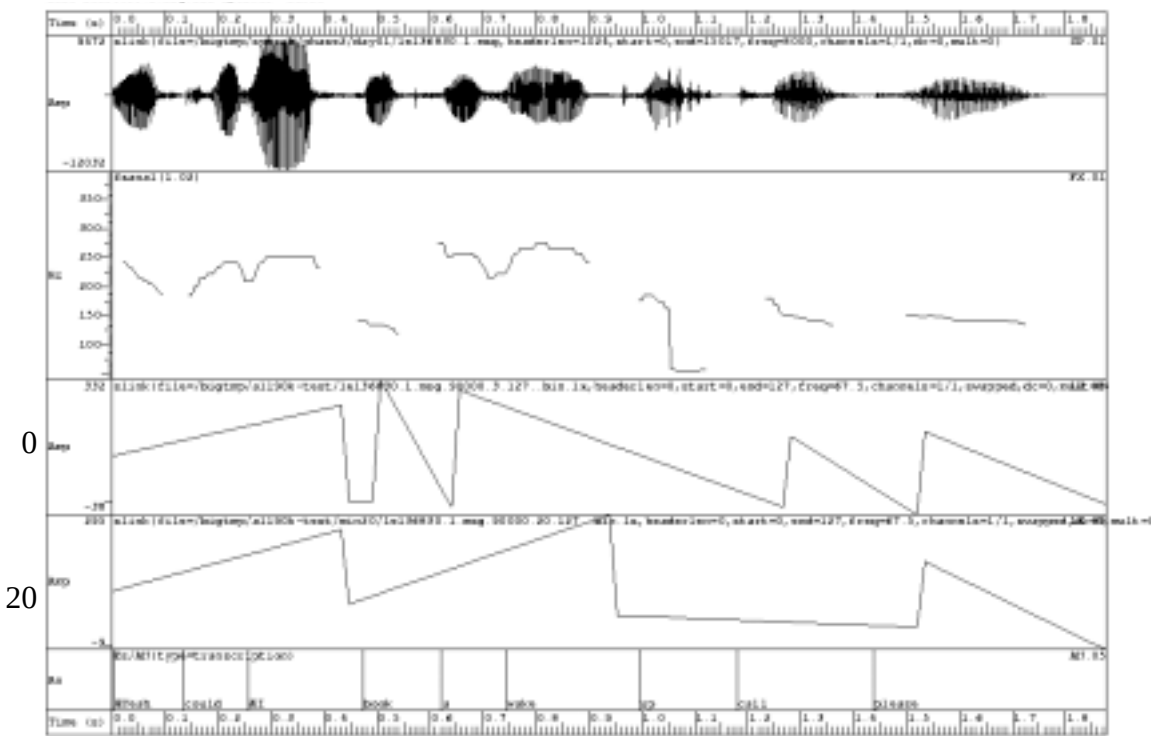

Figure 5.4 Segmentation of "Yeah could I book a wake-up call please?", with minimum segment lengths of 0 (that is, no minimum) and 20

5.1.1.2 Minimum segment length parameter

This parameter permits more explicit control over segment length than the segment insertion penalty; in many cases it works in tandem with the penalty to provide a more optimal segmentation.

The first segmentation of Figure 5.4 is not optimal because it includes sub-syllabic segments even given the segment insertion penalty of 90000. In “book”, the devoiced initial and voiceless final are the cause of two unwanted segments, one 0Hz, the other sharply falling. By imposing a minimum segment length of 20 sample points, as in the fourth pane of Figure 5.4, these micro-prosodic effects are eliminated, and the resulting segmentation is approximately "Yeah could I | book a wake | -up call | please?"

Another segmentation which is enhanced by the imposition of a minimum segment length is given in Figure 5.5. Without this treatment, two excerpts from the utterance, “Could you possibly” and “from Leicester”, which seem intuitively to require only one segment apiece, would be split into two parts.

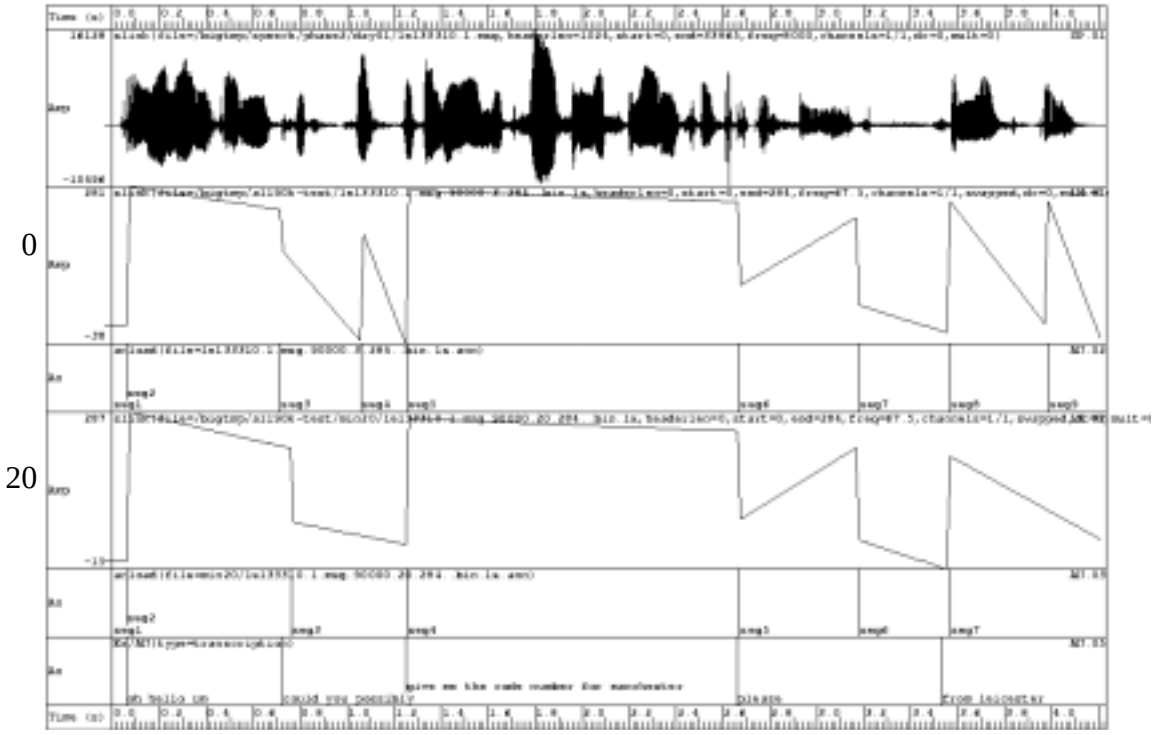

Figure 5.5 Another segmentation benefiting from minimum segment length

Occasionally, the imposition of minimum segment length has a deleterious effect. In Figure 5.6, segment 4 of the second pitch stylization consists of “...trying to get a number up in Chesterf[ield]”; segment 9 is “...off and on for about an hour and a half, and it’s always engaged, I wonder if you could...” In the first stylization of Figure 5.6, where no minimum length applies, these speech fragments are segmented into more manageable chunks.

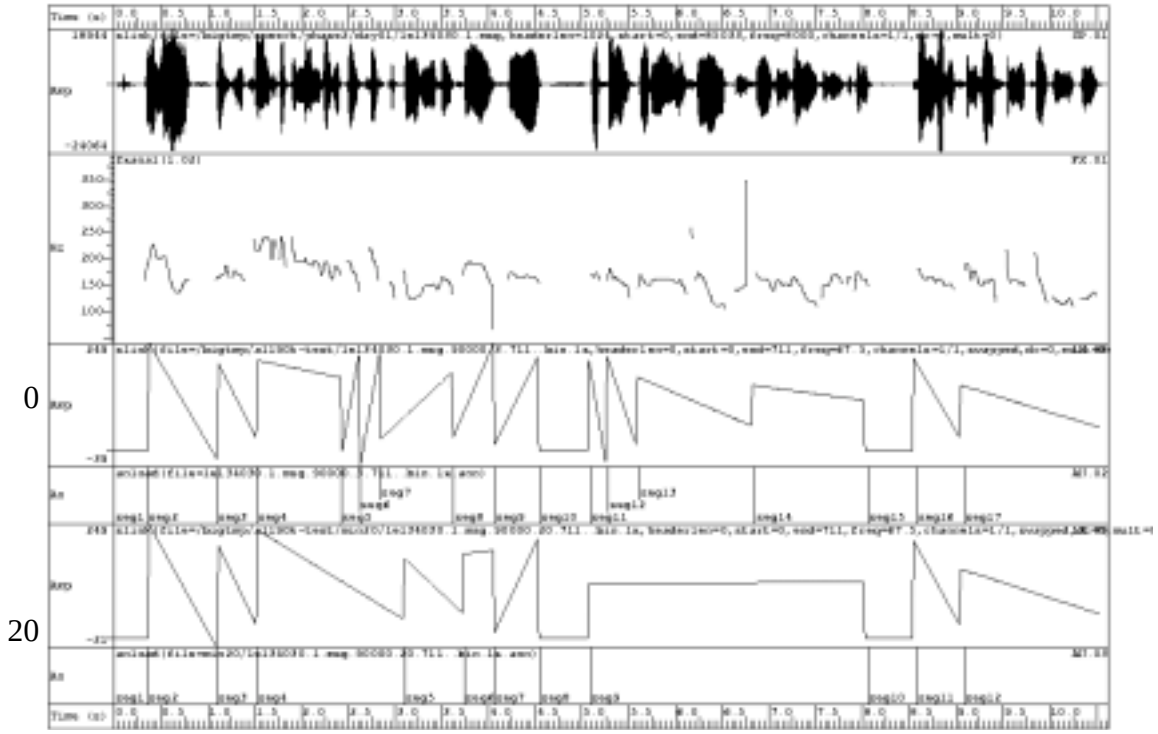

Figure 5.6 Utterance including overly long segments at minimum segment length 20. The stylization without minimum segment length is shown for reference.

5.1.1.3 Octave error correction

Inspection of Figure 5.6 also reveals octave errors – a halving error at 4 seconds, and doubling at 6.75 seconds. The integrated PDA attempts to cater for octave errors by assuming that most speech lies within a range of 50 to 500 Hz, and penalizing other behaviour.

An octave error correction algorithm implemented by David Moreno detects samples which are precisely double or half the frequency of the preceding sample, and in many cases yields an improved performance. Figure 5.7 shows further output from the utterance illustrated at Figure 5.6: the PDA output of the first pane may be compared with the octave-error corrected contour (actually a segmentation with fixed segment penalty zero and no minimum segment length) which lacks the two erroneous pitch excursions. Moreover, the second of the two overly long segments, noted above, is now divided into two more plausible segments.

Of a random sample of 35 utterances, close inspection revealed that minimum segment length improved segmentation on 17 occasions, had little impact 17 times, and had an adverse impact once. When both minimum segment length and octave correction were applied, the impact on performance was 13 improved, 16 not affected and 5 adversely affected. In this context, "improvement" consists of a segmentation which is visually less choppy, and more closely mapped to word or clause boundaries.

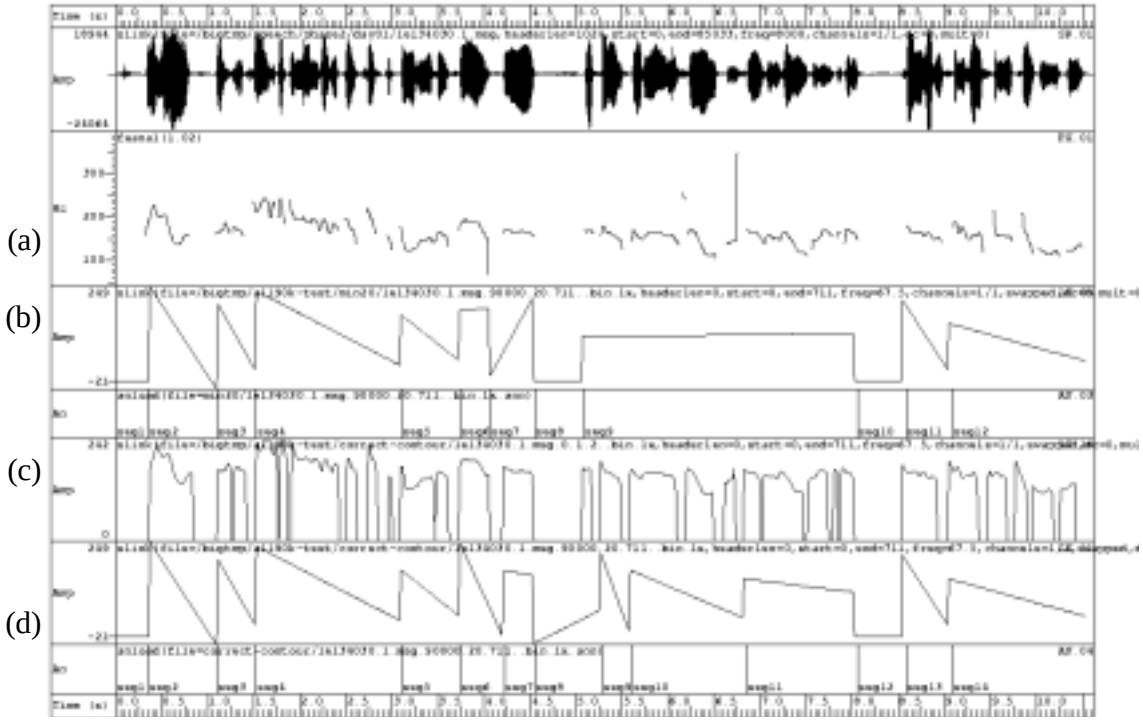

Figure 5.7 Exploiting octave error correction.

The F_0 trace and the trace processed by octave correction are shown at (a) and (c), and stylizations of each, using a segment insertion penalty of 90000 and minimum segment length of 20, appear at (b) and (d). Regions that the PDA has deemed voiceless are handled in one of three ways by the segmenter; Figure 5.8 includes an example of each. At around 0.3 and at 1.6 seconds,

voiceless regions occasion a segment boundary at a notional pitch of zero[2] or below, but they do not cause a voiceless segment – this we do however see near 0.6 and 1.2. The last couple of voiceless regions of the utterance, however, are not long enough to trigger a segment boundary, instead contributing to the declination pattern of the last segment.

Unvoiced segments have an important role to play in breaking utterances up into more manageable segments. This particular utterance asks for an area code, and the final segment consisting of "...for Reading please" would perhaps better have been subdivided. Some longer, sub-optimal segmentations, straddling voiceless regions are inevitable: this is the price paid for setting the fixed segment penalty sufficiently high that there are not too many infra-syllabic segments.

[2] Clearly, a fundamental frequency reading can never be less than zero. That a point on a stylized contour can fall below zero is of course a feature of the linear trajectory estimation.

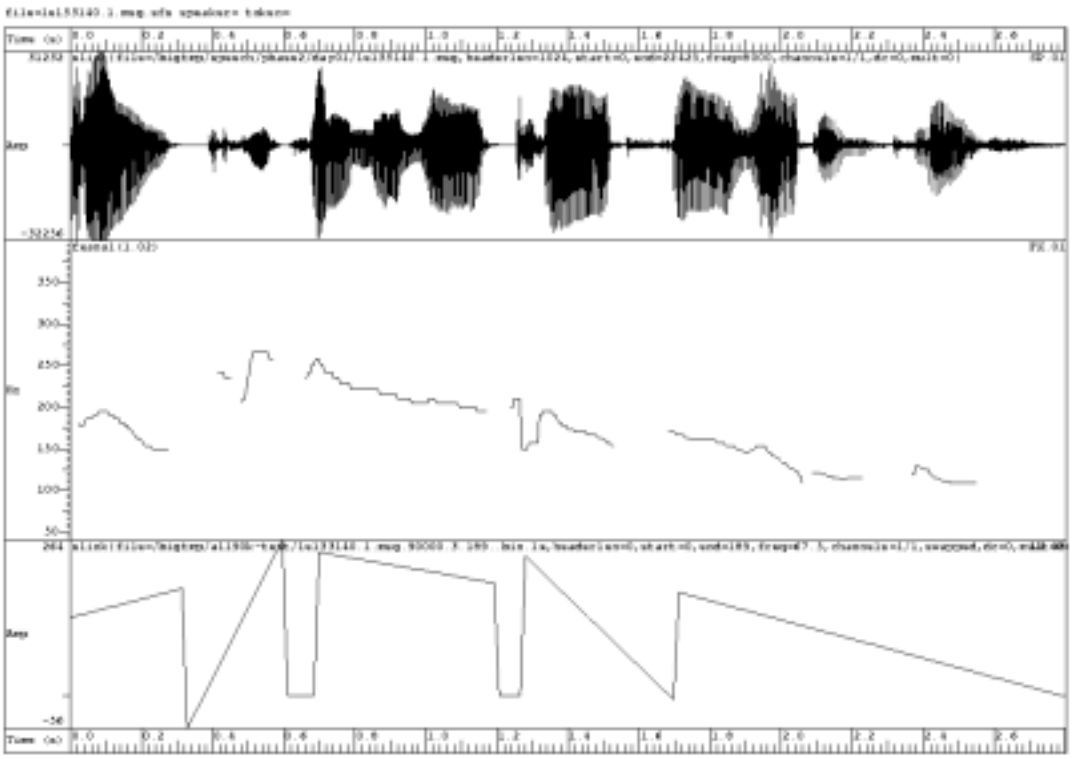

Figure 5.8 Segmenter behaviour in unvoiced regions: from the left, a segment boundary, two segments at a constant 0Hz, a further segment boundary, and finally a contribution from voicelessness to overall pitch declination

5.1.2 Clustering optimization

In this system, it will be recalled, the number of clusters corresponds to the number of available prosodic labels. In principle, this parameter should be set to whatever value yields the most successful classifications. In order to be of use as a labelling tool, though, the value should not be set too high, and ought to be of the same order of magnitude as that used in other labelling schemes. ToBI, for example, describes utterances in terms of a discrete set of intonational event types, which may have names such as *high rise* or *low fall*. A data-driven system such as PLoNQ is not subject to these phonological constraints; nevertheless, a set of centroids with a good mix of duration, average pitch and slope characteristics is necessary. A very small number of centroids is

therefore ruled out. Equally, if the parameter was set very high, there would not be enough examples of each centroid in the training data, and this sparse data problem would lead to poor classification results.

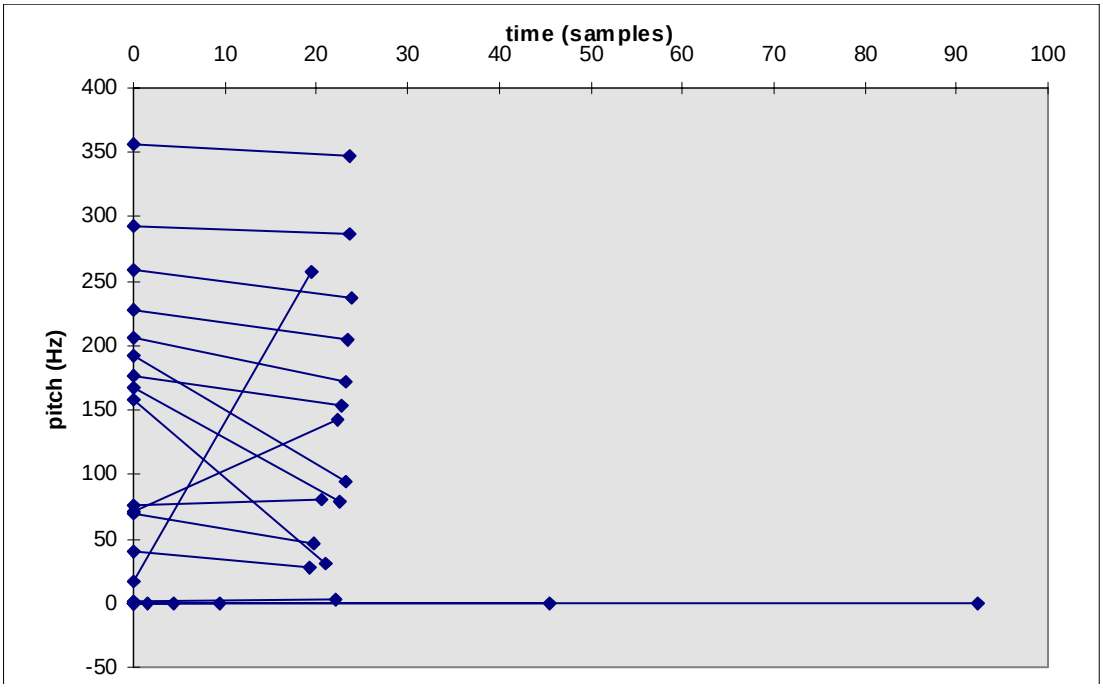

Figure 5.9 Centroid topology scheme – with maximum segment length 50

Figure 5.9 shows the characteristics of duration, average pitch and slope of twenty segment centroids; it is a schematic display of the prosodic label features. The centroid with the highest pitch, by way of illustration, begins just over 350 Hz and lasts for about 25 samples' duration, falling very slightly over the course of the segment, and is used by the classifier tool to describe that subset of training data segments whose features are better reflected by this than any of the other centroids. In the experiment run to produce Figure 5.9, the following parameter values were used:

pitch correction off

segment insertion penalty	0
maximum segment length	50
minimum segment length	20
number of centroids	20
normalization	off

Although there is some variation in pitch shape, with two of the centroid trajectories climbing, one steeply, and some others falling at various clines, many of them are flat and indeed only differ from each other in pitch level, with approximately equal duration. Only the five centroids with constant pitch at 0Hz depart significantly from this pattern in terms of duration, because they are not subject to the minimum and maximum length constraints, as has been noted. The reason that there are several such centroids is that, without pitch normalization, 0Hz segments represent unvoiced stretches in the training data, which are obviously more numerous than segments at any other given pitch.

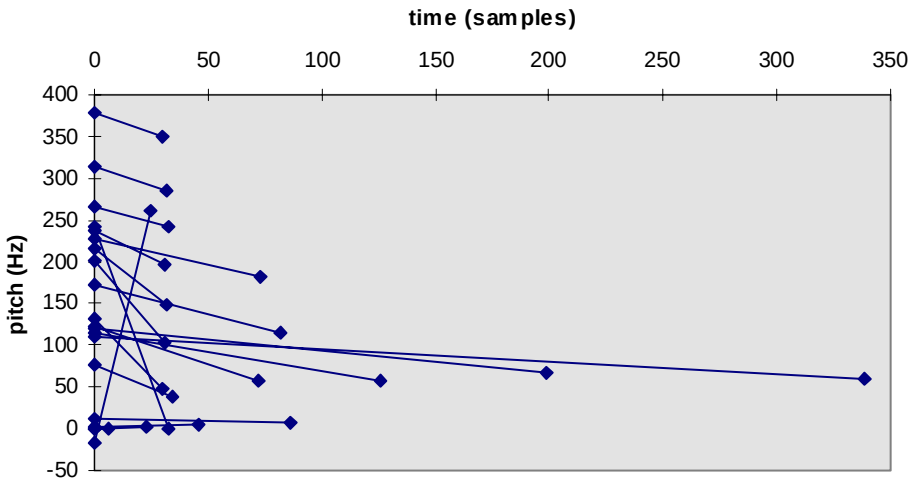

Figure 5.10 Centroid topology scheme – without maximum segment length

The only difference between the schemes shown in Figure 5.9 and Figure 5.10 is that the maximum segment length was applied to the former. Observe that relaxing this constraint has made available some longer centroid trajectories; their duration indicates that they cover sizeable

portions of the utterances in which they are found, so it comes as no surprise, given the phenomenon of downdrift (mentioned in Chapter 1), that they represent a gentle fall.

The descriptive power of the scheme of Figure 5.10 is marred somewhat by the loss of one of the rising centroid trajectories, however.

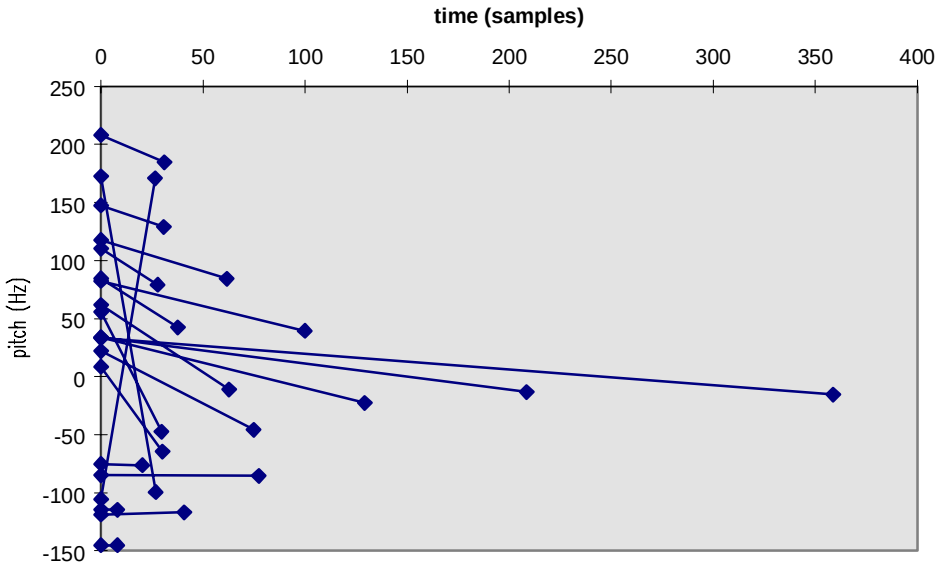

Figure 5.11 Centroid topology scheme – with normalization

Figure 5.11 shows the effect of normalization when applied to the scheme of Figure 5.10. Here, the dynamic range of speakers is taken into account, so that the average pitch (of each speaker) is set to zero. The overall dynamic range (of all speakers) has decreased, as expected, although to a limited extent, and there are no longer any flat contours at 0Hz, as unvoiced regions are now assigned a negative notional pitch which reflects each speaker's dynamic range. The two figures show clearly that this range does not vary enormously among speakers, because, it is surmised, unvoiced regions have a constant zero pitch in all utterances. The zero value is taken into account when computing average pitch, so that however high-pitched a speaker's voice, the minimum value is bound to zero.

The impact of normalization, therefore, seems to be to transpose pitch contours down by a hundred or so Hz, at the same time discarding potentially useful information: namely, the absence of voicing that a 0Hz value signals.

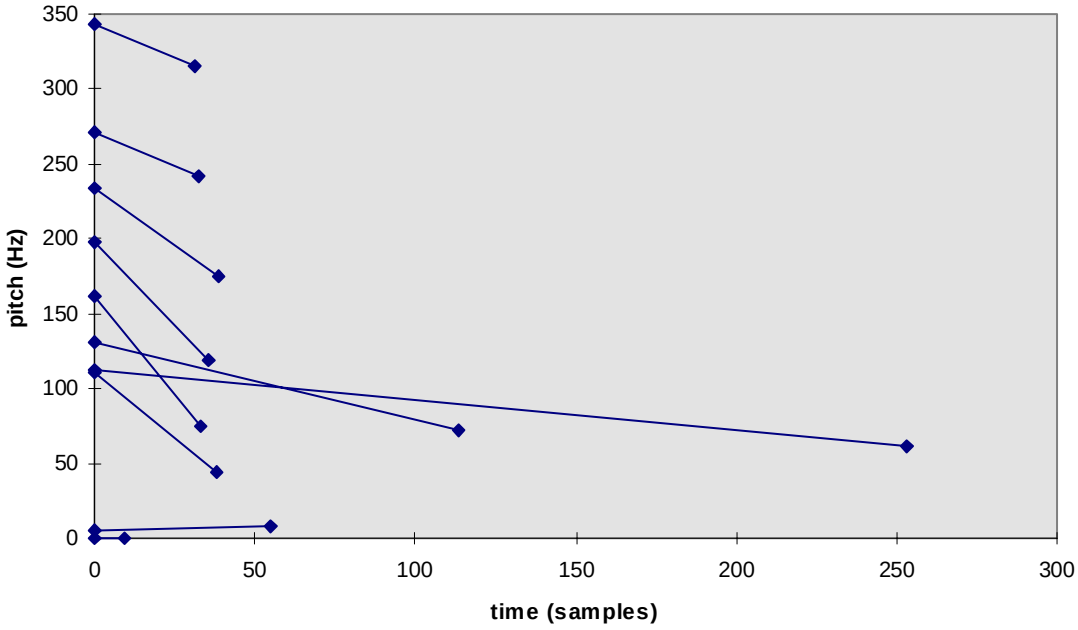

Figure 5.12 Centroid topology scheme – 10 centroids

Figure 5.12 illustrates the same scheme as Figure 5.10, except that the number of prosodic labels has been restricted to 10. As one would expect, the descriptive power of the Figure 5.12 scheme is poorer: rising contours, for example, are not in evidence at all. A scheme consisting of one or two label types would obviously be of no use at all in classification work; from this general trend we might conclude that the best scheme might consist of a very large number of centroids. Before we investigate this possibility, let us briefly look at the problem of centroid initialization.

5.1.2.1 Cluster seed files

In Chapter 4, when the K-means clustering algorithm was described, it was pointed out that each cluster must be initialized with arbitrary values for each feature. These values could be randomly generated; but for computational efficiency (minimizing the number of iterations of the algorithm)

it makes sense to choose values that are of the same order of magnitude as those that will be encountered in the training data. The simplest way to implement this idea is to take values from one of the training data files, and this is what was done in these experiments. The topology scheme of Figure 5.13 is the same as that of Figure 5.12, except that a different seed file was used. The choice of seed file appears to make very little difference to the scheme that emerges. This somewhat subjective judgement was not tested by exhaustive experimentation, however.

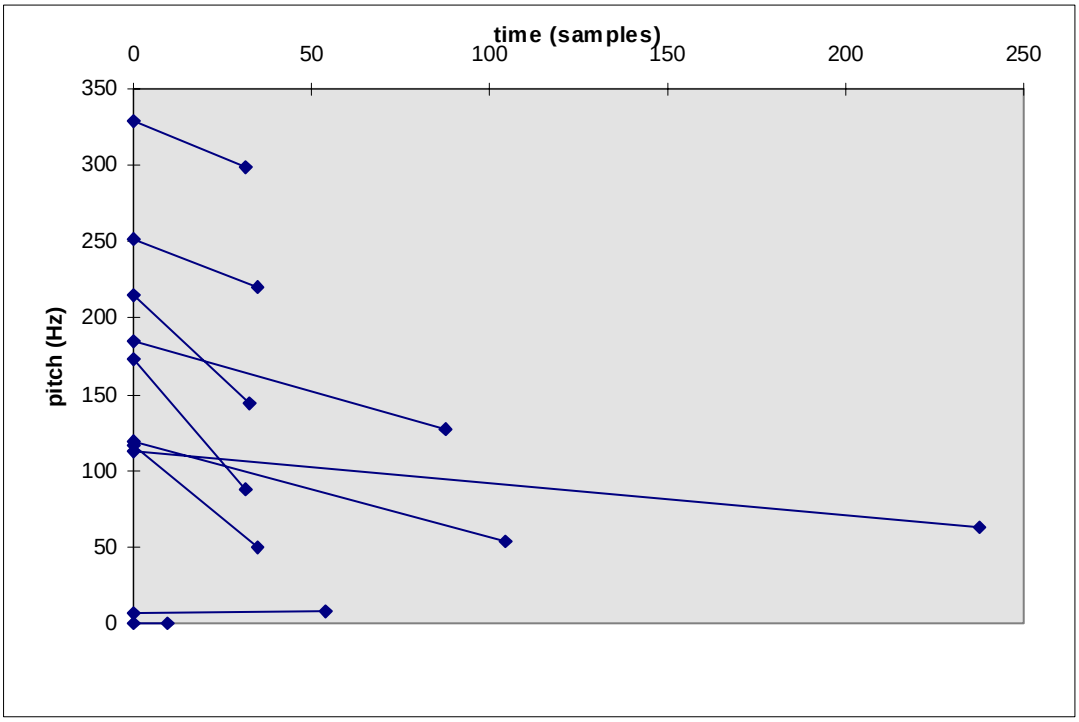

Figure 5.13 Centroid topology scheme – 10 centroids, different seed file

5.1.2.2 Raising the number of centroids

It can be safely assumed that a *clusnum* (number of centroids) of 10 is inadequate for our purposes, as no rising contours are available. Schemes with centroid order 20, it was demonstrated above, have slightly more descriptive power in that they incorporate one or two rising contours. It is instructive to compare Figure 5.10 with Figure 5.14 (whose time axis has been truncated for clarity): the latter attests rather more rising contours, but note that with one exception these additions are steep rises from zero, representing the onset of voicing. Their

profiles are so similar that one would not expect them to enhance the discriminative properties of the scheme.

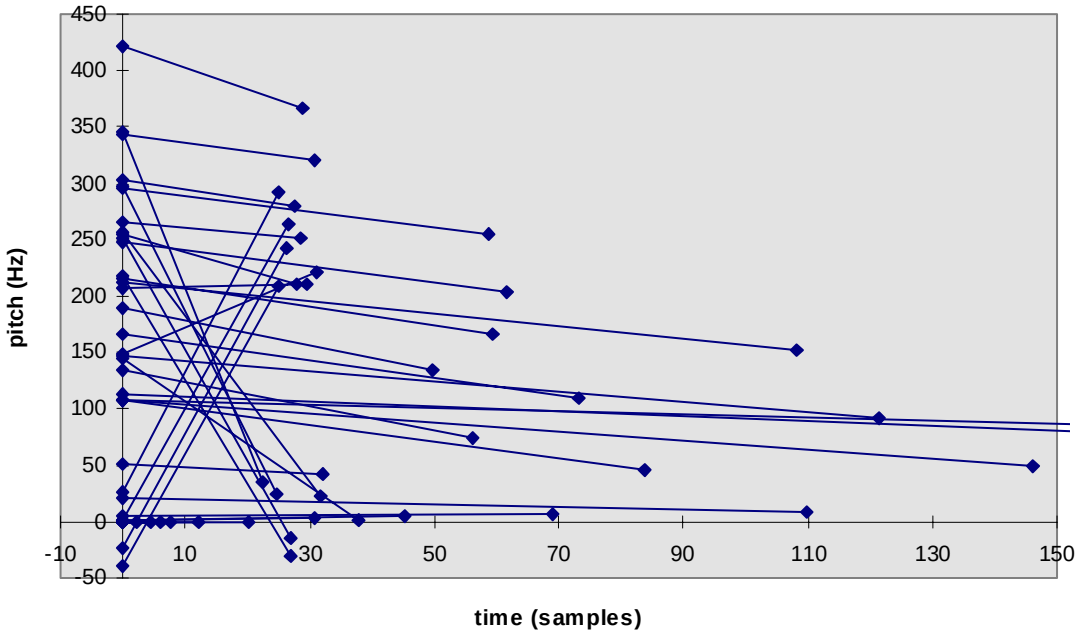

Figure 5.14 Centroid topology scheme with 40 centroids

With 70 centroids the trend continues, as shown at Figure 5.15. There is an explosion of similarly profiled contours, such that inspection by eye becomes tricky, but still only one rising contour of medium pitch and length.

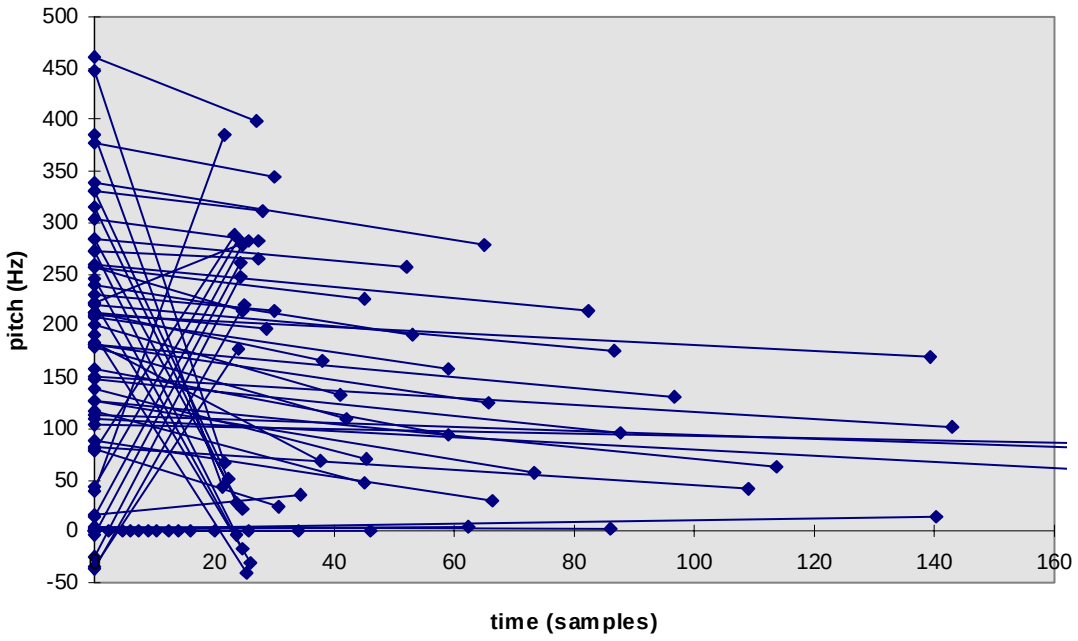

Figure 5.15 Centroid topology scheme with 70 centroids

This evidence indicates that no real benefit flows from setting the number of clusters very high, and that 30 or 40 is the optimal value.

5.1.2.3 Time-warped visual inspection of discretized contours

So far we have been considering centroid trajectories in isolation, attempting to assess their descriptive adequacy in terms of the variety of forms they take. It is equally important to look at examples of discretized contours in their entirety, to see how well they mirror the stylized contours they are meant to represent. Figure 5.16 sets out stylization and discretization data for the utterance “Hi can I have an early alarm call please?” The first pane is the speech waveform, the second the raw pitch contour, and below this the stylization, complete with segment boundary annotation. Underneath this are given discretizations with various instantiations of the cluster number parameter, namely 20, 30, 40, 50, 60 and 70, again with an annotation showing segment boundaries, and the number of the centroid assigned to that segment, given in the format s*segment number*/c*centroid number* (for example s1/c5). The centroid number is an arbitrary value, but is of

some interest where a particular centroid crops up twice in one utterance, for example.

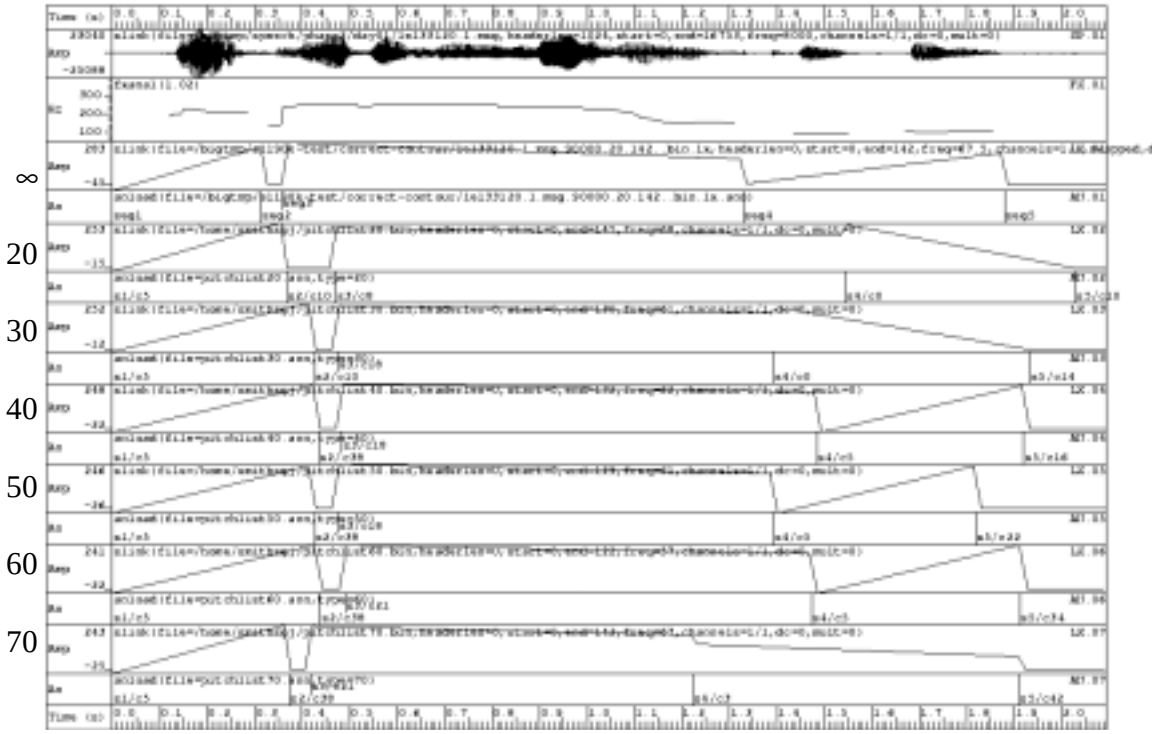

Figure 5.16 Discretized contours with different numbers of clusters. The symbol ∞ is used to indicate the (reference) stylized contour, which is effectively a discretization with no upper bound on the number of clusters available.

The raw pitch contour is given for guidance, but it is not of course the correlation of it and the stylized contour that is being evaluated at this point in the discussion.

The segments of the discretization are not of the same length as the stylized segments: while the latter may be of any duration, the former are assigned the features (including duration) of a cluster centroid. In the figures given here, therefore, the discretized contours are time-warped for ease of visual inspection, so that they fit into the same size pane as the other data. This means that the contour does not offer a comparative impression of overall discretization length (since the combined duration of discretized trajectories can, and almost always does, differ from that of the stylized trajectories, whilst the latter is by its nature equal to the utterance duration). For such an impression one can consult the *freq* parameter of the command shown in the relevant pane: the

closer it is to 67.5 (the sampling frequency of the PDA), the more accurate the modelled duration.

The time-warping also causes the slope of trajectories to be slightly distorted. The absolute pitch values are correct, however, as are the relative durations of segments within a pane, and the display gives a good impression of general modelling accuracy.

From Figure 5.16, it is apparent that the segment durations are best modelled with *clusnum* set to either 20 or 70, as the *freq* parameters turn out to be 68 and 67 respectively. In Section 5.1.2 above, it was suggested that neither of these values of *clusnum* is optimal, as 20 does not provide a rich enough set of descriptors, and at 70 there is a risk of overfitting the training data. Closer inspection reveals that although duration is modelled well at these parameter settings, the contour shape is not: for the fourth segment, an entirely inappropriate falling trajectory has been selected. This is the case, too, with *clusnum* set to 30. At first blush one might be surprised that the same trajectory as was used for the **first** segment was not selected, but it is duration and average pitch that have led to the selection of centroid 0 (centroid 3 for the case where *clusnum* is 70).

The general paucity of rising trajectories, especially among labelling schemes where *clusnum* is low, has already been noted. Where no trajectory of appropriate gradient is available, the system will simply make a selection based on the other two prosodic features. In the next example, Figure 5.17, there is no *clusnum* setting that adequately models the first segment, and the most faithful discretization of the second segment, where *clusnum* is 40, is acceptable on contour shape but not on duration.

In none of the discretizations has there been any difficulty finding a trajectory to represent the unusually long final segment; recall that two or three examples of this sort of trajectory were available in all the centroid topologies presented above.

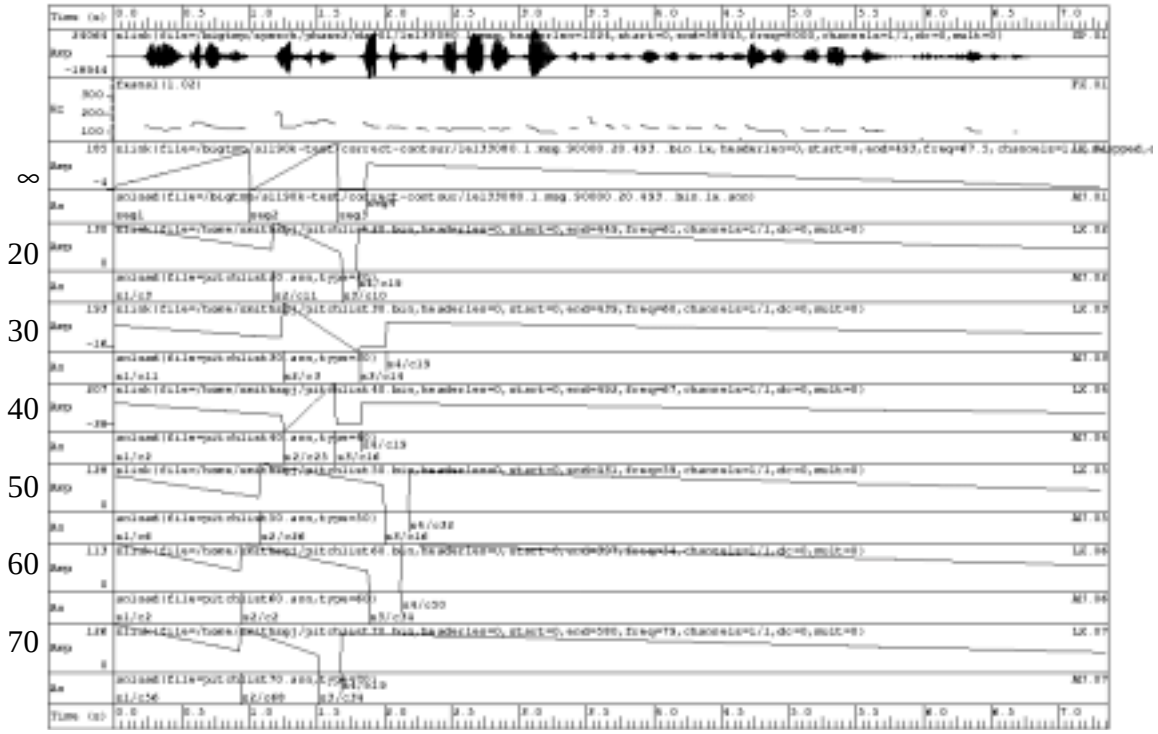

Figure 5.17 Discretization with poorly modelled gradient. The first segment of the stylized contour (∞) is not well modelled by any of the *clusnum* schemes; the second is reasonably well modelled only by the discretization where *clusnum*=40.

A final example demonstrates what ought in any case to come as no surprise, given the durational variety of 0Hz trajectories found in the centroid topologies above: unvoiced segments, such as segments 1 and 3 in Figure 5.18 are in general quite well modelled. Once again, the discretization with *clusnum* set to 20 looks impressive, but the duration (*freq* is only 55) belies this. In this case the optimal discretization seems to be at *clusnum* 70, but performance at 40 and 50 is also quite adequate, and it will be recalled from Figure 5.16 that setting this parameter very high sometimes leads to deterioration in performance.

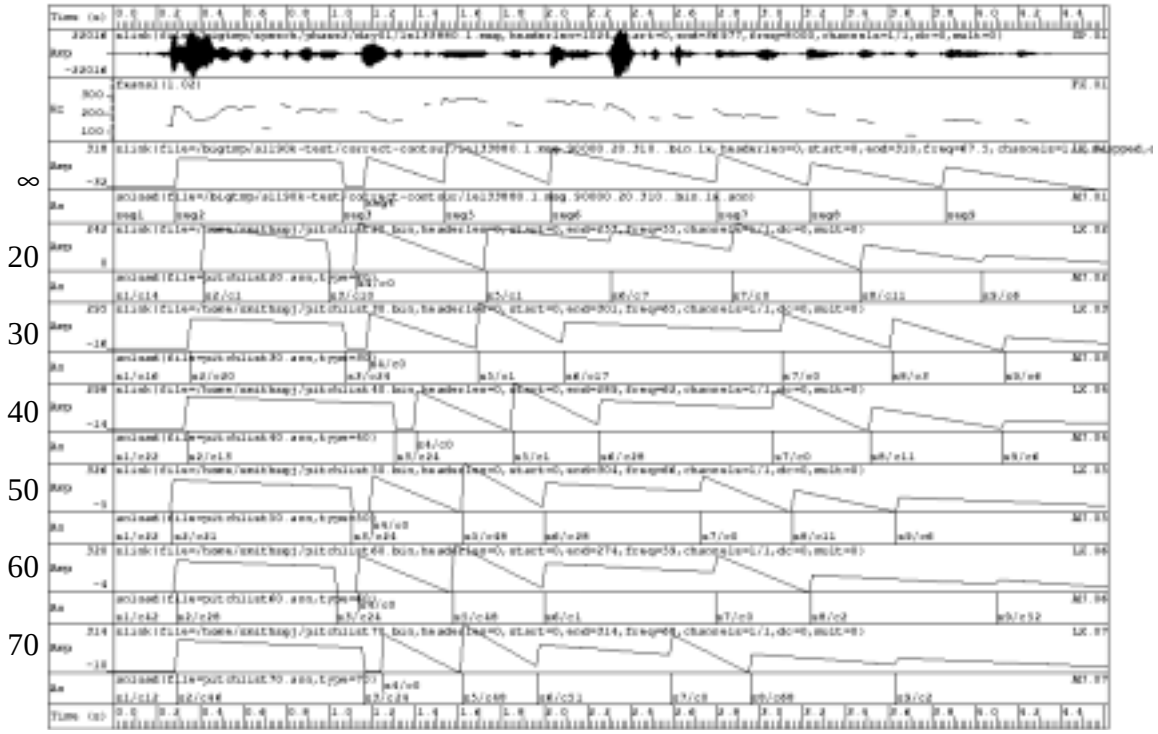

Figure 5.18 Discretization with two unvoiced segments. At all values of *clusnum*, the two voiceless segments (the first and third) are quite well modelled.

Assessment of useful values for the *clusnum* parameter has so far taken a somewhat impressionistic and perhaps subjective approach. In the next section we turn to a more quantitative analysis.

5.1.2.4 Cluster variance, and secondary training data

Variance is the measure of the average distance between a cluster centroid and the data points assigned to that cluster; it helps us to visualize how well a prosodic label can model the features of the prosodic segments assigned to it during clustering.

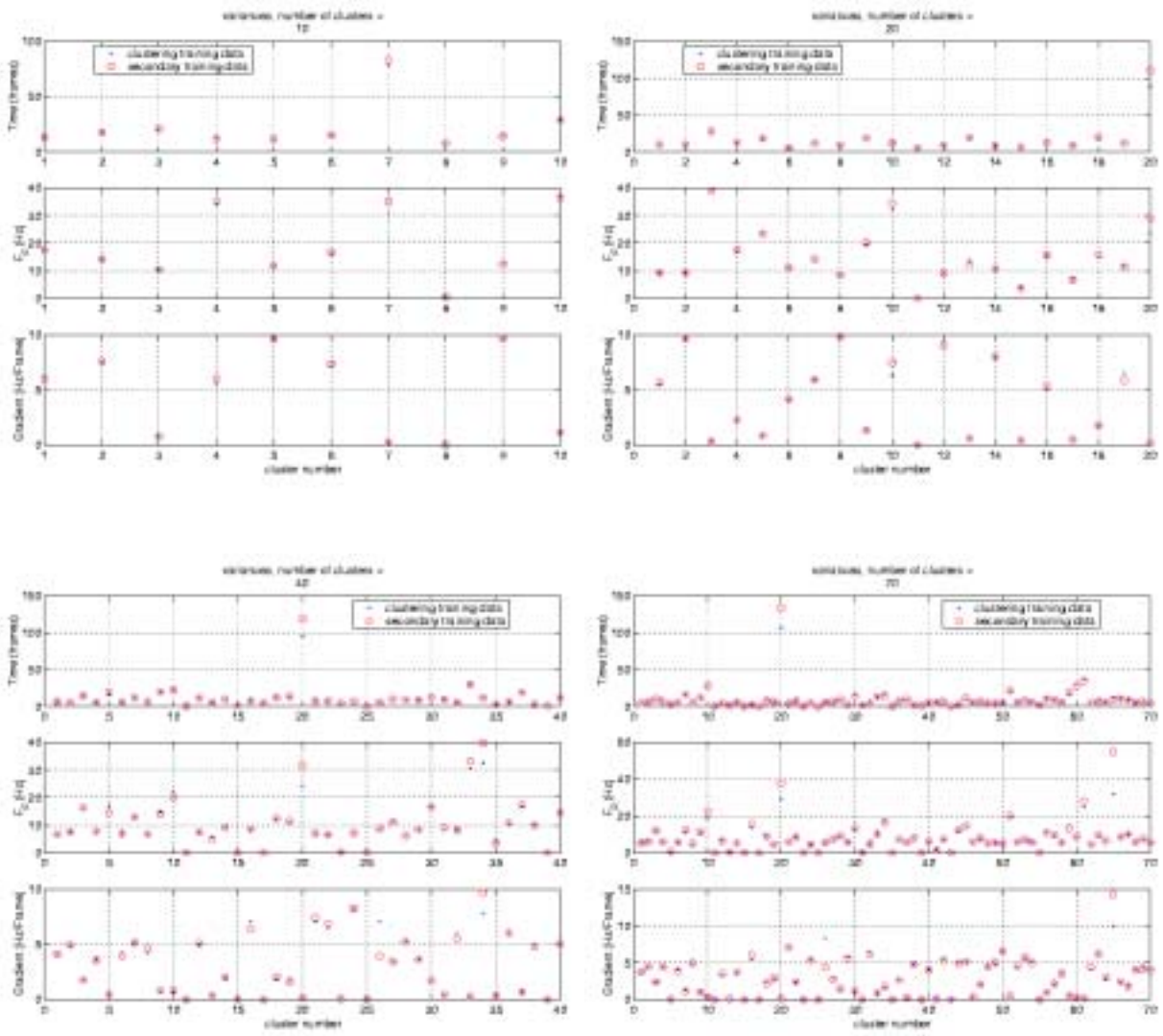

Figure 5.19 Variances between centroids and cluster members, at four different *clusnum* settings

Figure 5.19 is divided into four windows, representing variance results at four different values of *clusnum*. The three panes within each window give data for the three features comprising the prosodic segment feature vectors. For the centroids in each *clusnum* scheme, there are two data points: the dots (many of which are located in circles) shows the variance between the centroid and the data used to derive it. Each circle indicates the variance between the trained centroids and a secondary set of data labelled according to them, in exactly the same way as test data would be

labelled. It will be recalled from Section 5 of the last chapter that a secondary training stage was considered, in order to fend off the risk of data overfitting. It is apparent from Figure 5.19, though, that the variances with respect to primary and secondary training data differ little (the distortion between them is low), with the exception of one or two data points per scheme which are in any event outliers. The distortion grows slightly with the increase in the value of *clusnum*, as one would expect, for distortion would be maximized given a number of clusters equal to the number of segments in the data, but is never significant. This is quite an important finding, for it demonstrates that the prosodic labels generated by the segmentation and clustering procedures are of sufficient descriptive power to capture most features of unseen data.

It will come as no surprise that as *clusnum* is increased, variance diminishes: by extension, in a putative scheme where the number of clusters equalled the number of segments, variance would clearly be zero in all cases. Figure 5.19 confirms this expectation, with data points congregating near the bottom of the value axes as with greater numbers of centroids. Even at 70, however, there are still distant outliers; observe that under this scheme the tool used to make the graphs has changed the scale of the value axis to accommodate one such outlier.

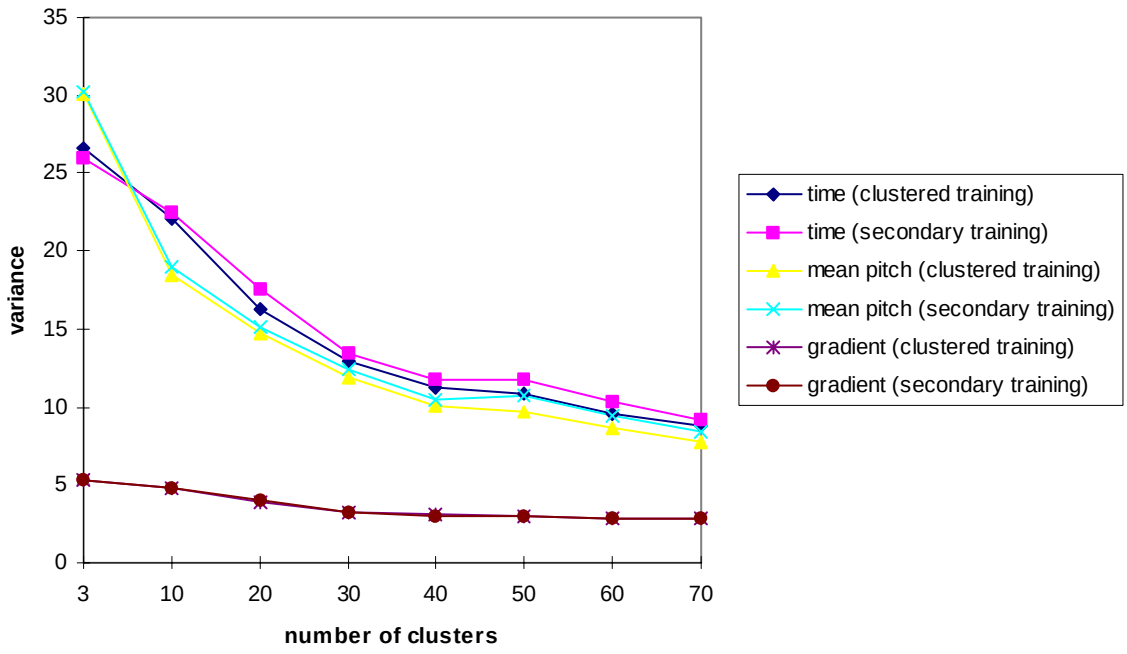

Figure 5.20 Average variance of the three features, in both training sets, under various *clusnum* schemes

Figure 5.20 confirms that the two trends apply globally. Variance for gradient appears identical over the two training sets, so that overall distortion is close to the ideal value of 1. For the other two features, time and mean pitch, the two curves track each other closely. The trend for variance to decrease as *clusnum* is increased is also evident.

Distortion is charted in Figure 5.21, for the same four *clusnum* schemes presented in Figure 5.19. The trend for distortion to increase with *clusnum* may be observed here.

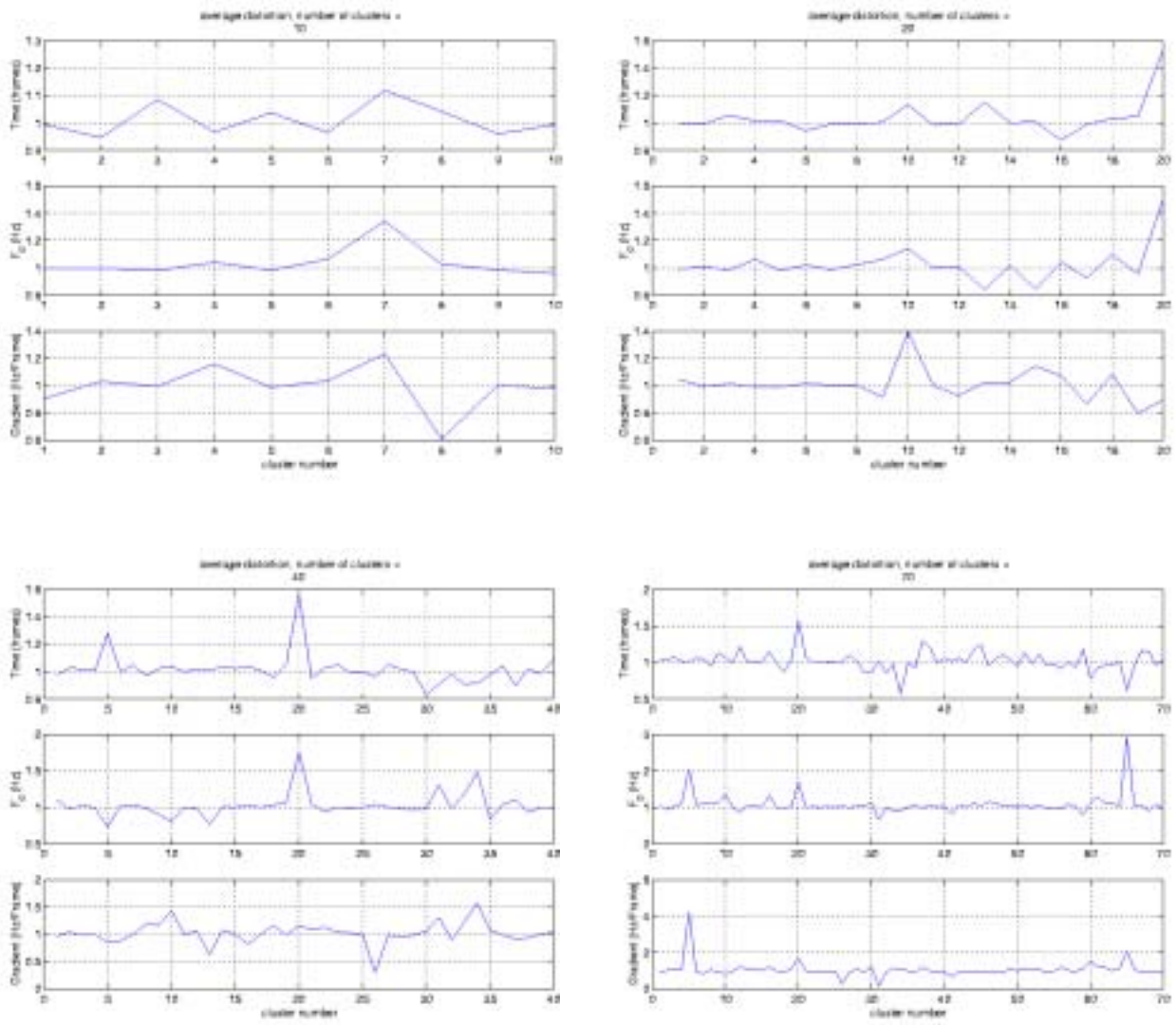

Figure 5.21 Distortion of primary and secondary data variance, various *clusnum* settings

The plots in Figure 5.22 are intended to give an impressionistic view of the distribution of centroids in three-dimensional space.

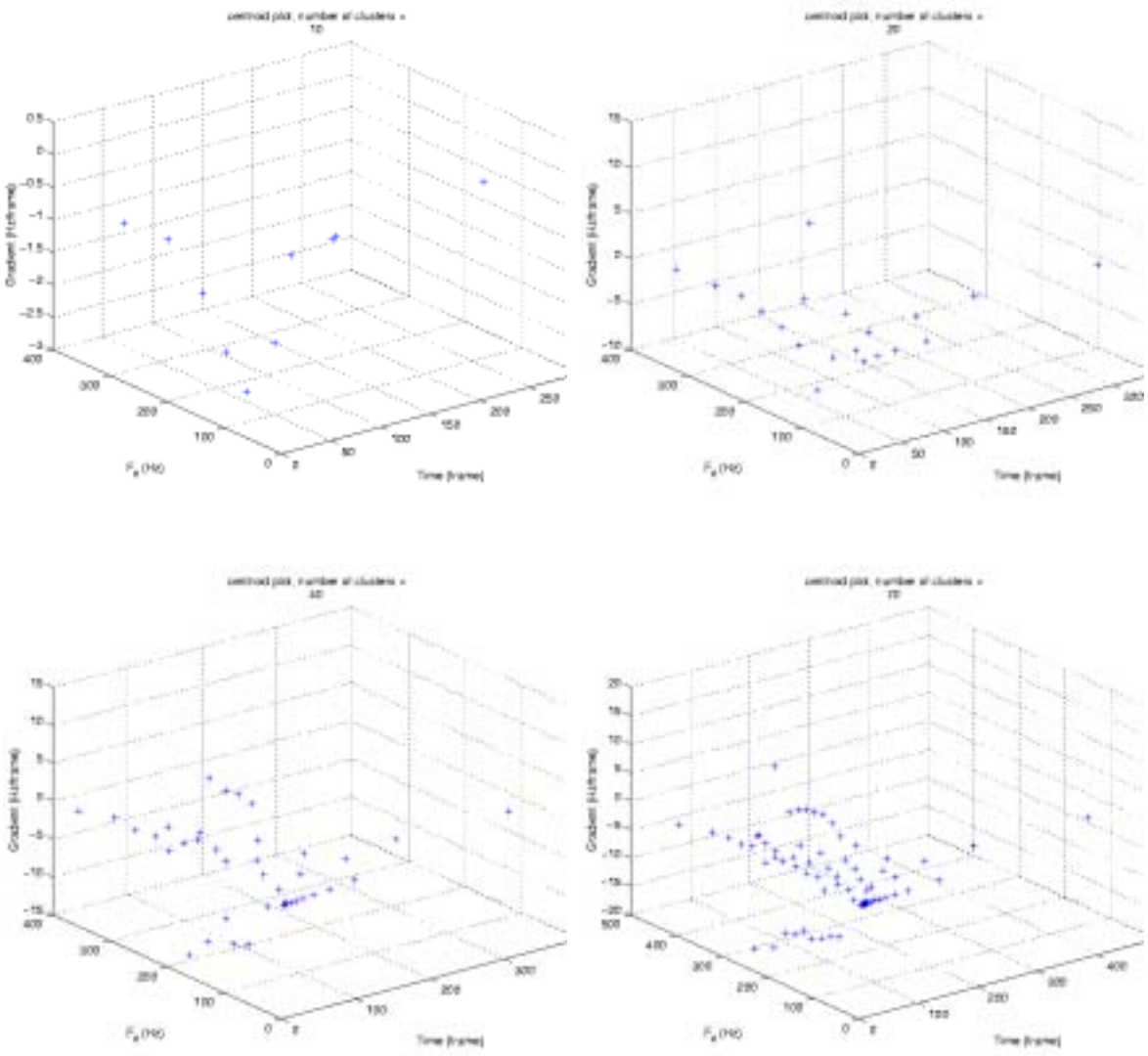

Figure 5.22 Centroid plots in 3-dimensional space

5.2 Test parameters

It was mentioned at the beginning of the chapter that because of the great scope for parameter adjustment throughout the classifier suite, it is necessary to navigate a critical path through the system, finding optimal parameter settings along the way. Failure to do so would result in a combinatorial explosion of potential approaches to classification, and an intractable number of experiments to be performed.

This sort of intermediate tuning and validation is what has been attempted in the foregoing sections. It was shown, for example, that a segment insertion penalty of 90000 was often optimal, because it tends to provides better modelling of the F_0 contour. Likely optimum values for *clusnum* were similarly motivated, and these will be verified empirically in the remaining sections of this chapter.

The parameters which have already been discussed in this chapter – given in Table 1 – were defined as training parameters. The other parameters – those described from this point on – are known as test parameters. They are:

1) *n-gram order* (of prosodic label sequences)
2) *absfreq* (absolute frequency of n-gram in training data
3) *MI* (mutual information of last label in n-gram and its antecedents)
4) *salience* (of n-gram, to some class, in training data)
5) *floor value* (estimated probability of n-gram not found in training data)
6) *wprior* (a weighting constant which determines the influence of the prior probability of each class in determining overall probability)

5.3 Test data

It was explained in Chapter 3 that the Oasis corpus is arranged in groups of directories according to when the recordings were made; and that for experimental purposes it is orthogonally divided into nine segments, which each draw on all of the recording sessions. In this work, segments 7 and 8, consisting in total of 1989 utterances, were used as test data. There were two sets of training data, as mentioned previously: the primary (clustering), and the secondary (prosodic label sequence modelling) training data. Sets consisting of segments 1, 2 and 3 (2995 utterances) and segments 4, 5 and 6 (2987 utterances) were used, respectively, for these purposes. Each segment in the corpus, except segment 9, which is not used at this stage, consists of 1000 utterances, but some of the utterances had to be discarded because they were too long for the segmentation algorithm to handle, as noted in Chapter 3: the time required to segment an utterance increases

exponentially with its duration, and the 41 discarded utterances would have taken many hours each to process.

As Oasis is comprised of six classes, if one were to classify each utterance in the test set at random, one would expect to make 331.5 correct classifications, corresponding to a probability of 0.17. This does not, however, take into account class prior information. As the most populous class, *info*, is represented by 597 utterances in the test set, correct classification by chance amounts to 597÷1989, that is 0.3.

In order to demonstrate that the algorithms were working correctly, and were not trained for optimal performance with a particular set of test data, an experiment was conducted where the data sets used for initial and secondary training, and testing, were shuffled, with the expectation that the results would be about the same as with the main test-training configuration. There was also a cheating experiment, where secondary training data was used as test data; here, it was expected that the results would be significantly better than the mainstream experimentation.

5.4 Prosodic label sequences

The first set of experiments looked into the utility of prosodic label sequences (bigrams, trigrams and so on) for classification. For the present, the other test parameters are left at default values. N-gram orders of unigrams only (1:1), unigrams and bigrams (1:2), unigrams, bigrams and trigrams (1:3), and so on up to eight-grams (1:8) are considered; in Figure 5.23, the n-gram order is plotted against the number of successful classifications (out of 1989).

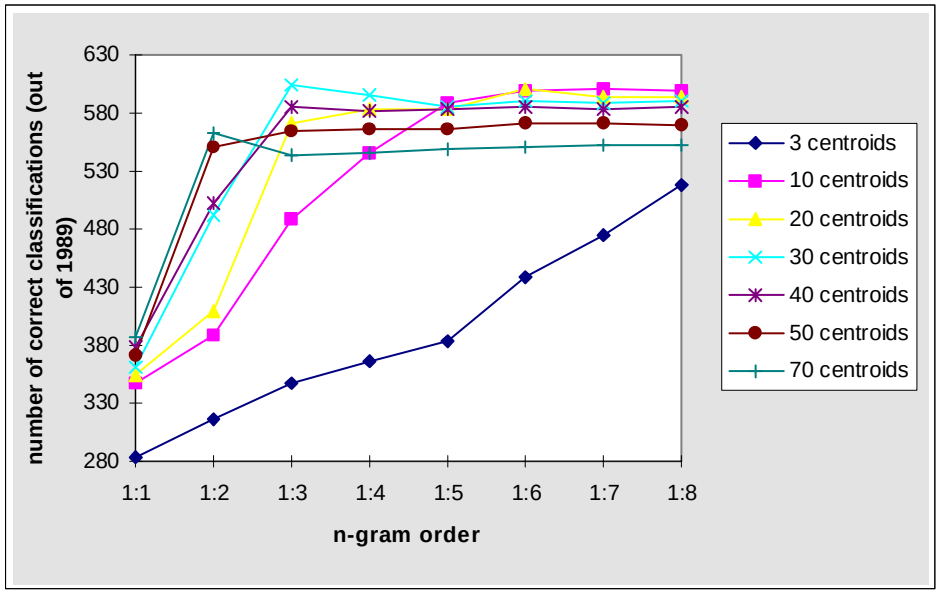

Figure 5.23 Classification performance under various n-gram orders and labelling schemes

The worst performance, as expected, occurs when only unigrams are used: this applies whichever labelling scheme is used (that is, whatever value *clusnum* is set to). The best performance is with a *clusnum* of 30 and an n-gram order of 1:3, with 605 correct classifications (30.42%), followed closely by a score of 595 (29.91%) with the same n-gram order, and *clusnum* of 40. This result is gratifying, in that it mirrors the impressionistic evidence of the time-warped discretization diagrams presented in Section 5.1.2.3. It was found, it will be recalled, that an available 30 or 40 prosodic labels appeared to provide closer representations of the stylized contour than was the case with other schemes.

The peak performance at 1:3 does not apply with other *clusnum* schemes. Observe, in particular, the trend whereby the higher *clusnum* is set, the lower the n-gram order at which the peak is reached. The 10 centroid scheme produces quite good results, with 599 correct classifications (30.12%) at 1:6, but it is computationally more efficient to adopt a scheme with a lower number of centroids.

The fact that the performance of each *clusnum* scheme peaks at a given n-gram order and then decays may be explained as follows. Where n-gram order is very low, sub-optimal performance is to be expected because the distribution of label sequences is arbitrary. In a unigrams-only scheme, for example, there are only *clusnum* "sequences" available. Each one will almost certainly be attested in utterances of all classes, and therefore make no discriminative contribution. With a very high n-gram order, on the other hand, a large number of sequences will be found in the test data which do not occur in the training data, making no contribution to the predictive power of the scheme. This in itself does not affect performance, as such sequences are ignored in the likelihood calculations, and for that reason we might be led to expect performance plateaux, rather than peaks, on the n-gram order axis of Figure 5.23. The performance curves do not, in fact, drop off particularly sharply; the reason that there are peaks at all is that sometimes a longer n-gram is indeed present in the training data, but it is too rare to be a reliable predictor of class membership. Paradoxically it is when longer n-grams *do* co-occur in the test and training sets that performance is adversely affected: the rarer they are, the lower the chance that they properly represent the same UT, and misclassification is the inevitable consequence. When the n-gram is not found in the training data it is, as stated, simply disregarded.

The second finding, that the n-gram order performance peak is reached more quickly when *clusnum* is high, also flows from this sparse data problem. With a higher n-gram order, the large number of possible sequences, and consequently the small number of occurrences of any particular sequence, means that the statistics available are somewhat unreliable.

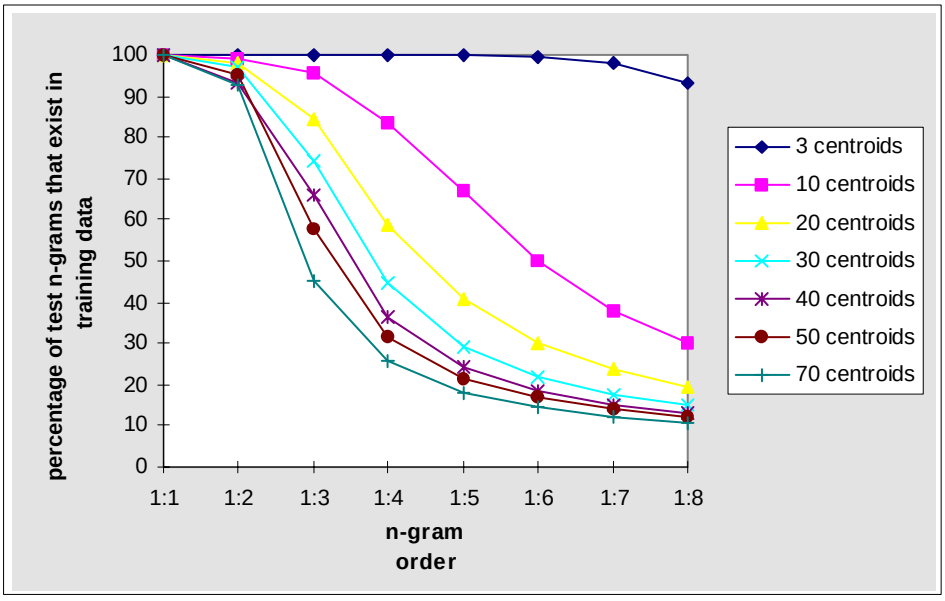

Figure 5.24 Proportion of n-grams encountered in test data that are present in training data, at various n-gram orders

Figure 5.24 confirms that at a low n-gram order, prosodic label sequences are more likely to co-occur in the test and training data, and that the proportion of n-grams used dwindles a good deal more rapidly when *clusnum* is set high. One might surmise that a model which makes use of less than half the unseen sequences available is unlikely to perform well, as is the case for the 70 centroid model with an n-gram order above 1:2. Maximal use of the available data, tempered by sufficient context and discriminative power, assures the best performance: this accounts for the success of the 30 centroid model at n-gram order 1:3.

5.5 Validation of experimental results

Some checks and balances are presented at this point, in order to confirm the consistency of data sets, and the reliability of results.

5.5.1 Data set shuffling

First, part of the experiment tabulated in Figure 5.23 is repeated, but with different corpus segments representing the test and training sets. Data sets were shuffled, so that the initial (clustering) training set, secondary training set and test sets were of different composition to that of the original experiment. Furthermore, the segments making up each set were not simply reassigned *en bloc*, as far as possible: for example, segments 1, 2 and 3, the original initial training set, were divided between the secondary training set and the test set. This experiment was limited to 30 and 40 centroid models, with n-gram order up to 1:6.

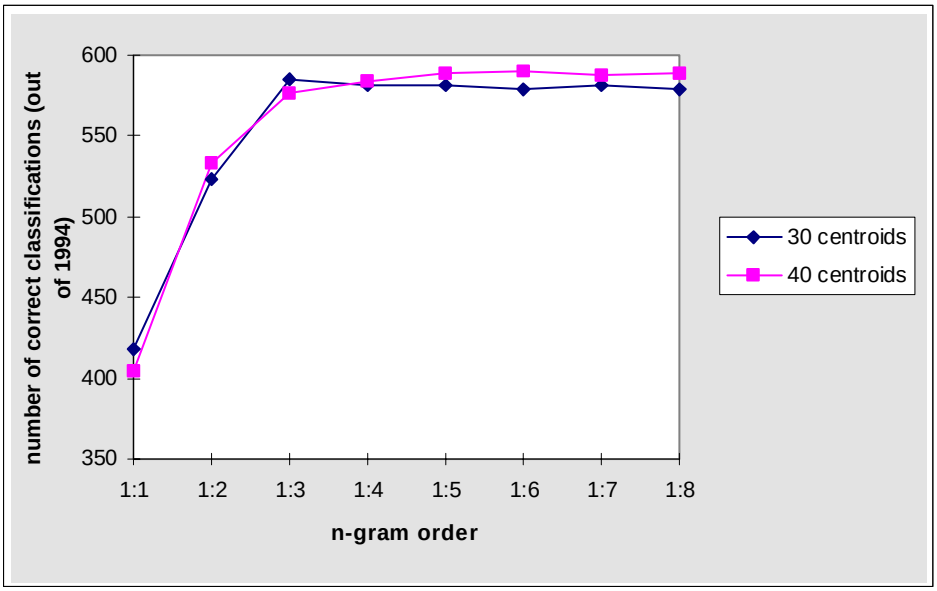

Figure 5.25 Performance with shuffled training and test sets

Figure 5.25 shows broadly the same results as the earlier experiment. The best performance with *clusnum* at 30, is again obtained with an n-gram order of 1:3; the successful classification score is 585 (29.41%), slightly lower than previously. The 40 centroid model peaks later, at 1:6, as predicted by the analysis above, with a successful classification score of 590 (29.66%).

5.5.2 Cheating experiment

A further validation experiment was concerned with algorithm verification. A cheating experiment was conducted, where the secondary training data was treated as test data, with the expectation that performance would be markedly better than under conditions using genuine test data; and that performance would not peak at a given n-gram order, but would continue to improve as the n-gram order was increased (because there would be no point at which "unseen" prosodic label sequences were not found in the training set, evidently). These expectations were confirmed, as shown in Figure 5.26. Any other finding would probably have pointed to a programming error.

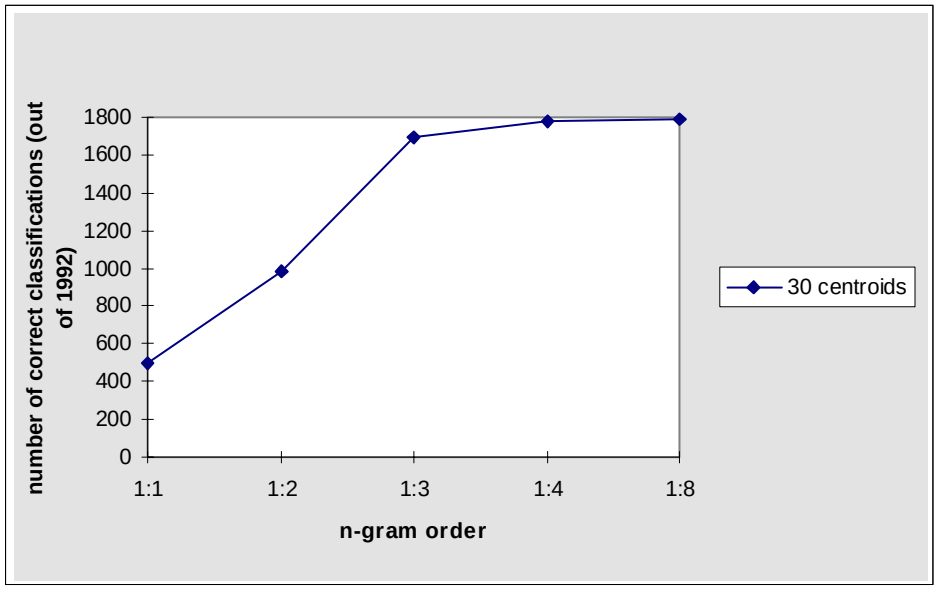

Figure 5.26 Classification results in the cheating experiment, showing the number of correct classifications at different n-gram orders, with *clusnum*=30.

As the n-gram order is increased, performance indeed improves; but the rate of improvement gradually becomes vanishingly small. Notice that the rightmost n-gram order increment shown in Figure 5.26 is from 1:4 to 1:8. Correct classification of all 1992 utterances could never be attained, even if very high-order n-grams were incorporated into the model, because the contribution of the lower-order n-grams is always taken into account.

5.5.3 Random data experiment

An experiment using random data instead of genuine training data was then performed. Training data input to the classification program consists of a text file listing training data segments and details about them, including the centroid number of the prosodic label assigned. This column was exchanged for a column of integers in the same range from a random number generator. This experiment was again conducted with a *clusnum* value of 30. The generator output was checked to ensure that approximately equal numbers of all 30 integers were issued.

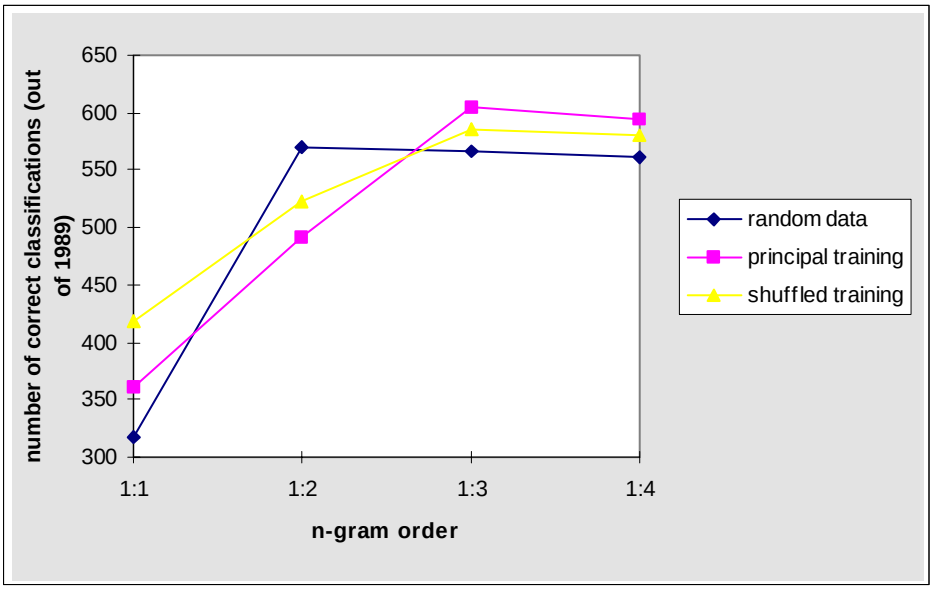

Figure 5.27 Comparative performance with random and non-random training data

Considering Figure 5.27, the reader will probably have two, related, concerns about the results reported: first, that the random data curve is quite close to the training data curves, and indeed at one point shows that the random model has outperformed the others. The second concern will be that the random curve rises, peaks at n-gram order 1:2, before beginning to converge with the training curves at 1:3, whereas one might have expected it to remain reasonably flat at around 330 correct classifications, or about one sixth of the total. The actual random performance at 1:3 is 570 (28.66 %).

The correct classification rates mirror, to some extent, the representation of each UT class in the

corpus. At n-gram order 1:1, whether real or random training data is used, the result is indeed random, as every label type occurs in all classes. At n-gram orders 1:2 and above, there are some sequences which do not occur in every UT class: when this happens, they will be assigned probabilistically to a class where they *do* occur. Over the course of a large experiment, it follows that the classification will tend to follow the distribution of classes in the corpus, for if *info*, say, is the most populous class, more unseen utterances will be deemed by the classifier to belong to that class; and the classifier will, naturally enough, be right.

This phenomenon applies even though the class prior parameter has not been activated at this point, and continues to hold as the n-gram order is increased; in fact, with each increment, the results are skewed a little further by the phenomenon, as more and more n-grams are encountered in test data which are absent from the training data. However, because of the more complete descriptive power of the 30 centroid model at 1:3 and 1:4, we can expect correct classification at significantly above the findings for random data.

A glance at Figure 5.27 indicates that, in fact, the non-random results are not spectacularly better than the random, even at these operating points. Therefore, a subsidiary experiment was run, in which 8 further sets of random data were generated for the 30 centroid case, and test utterances classified at 1:3. The results are presented in Table 5.3.

Table 5.3 Numbers of correct classifications in further random classification experiments

experiment no.	Successes
1	547
2	559
3	577
4	558
5	568
6	564
7	559
8	533

The average classification rate is 558.125. This is noticeably below the successful classification rate for either the principal (605 correct) or shuffled (585 correct). Furthermore, none of the experiments attained these levels of performance (although Experiment 3 came fairly close); it is

therefore concluded that the principal experiment of this section does describe random behaviour.

This is fortunate, of course, as we would have to infer from the opposite conclusion that the segmentation and labelling work reported in the thesis thus far serves only to generate a sequence of numbers as arbitrary as those provided by a random number generator.

5.6 Omitting secondary training

It was argued in Chapter 4 that the classification should be informed by two sets of training data: the initial training set underwent segmentation, followed by a clustering process which assigned prosodic labels to each segment, while the secondary set was segmented, labelled according to the features of the closest centroid in the initial set, and then used to determine the sequences of labels most closely associated with the various classes. By decoupling the definitional and assignment phases of the labelling task, it was claimed, the likelihood of overfitting of training data was reduced.

Evidence presented in section 5.1.2.4, however, seems to militate against this reasoning. It was shown in Figure 5.19, for example, that the variance of segment features from features of the associated label was approximately the same in the case of the two training sets.

A comparative experiment, with and without secondary training, was carried out; suppressing secondary training involved building the text file referred to in the last section (one of the inputs to the classifier module) from the utterances, segmentation and labelling of the initial training set. Figure 5.28 gives the results.

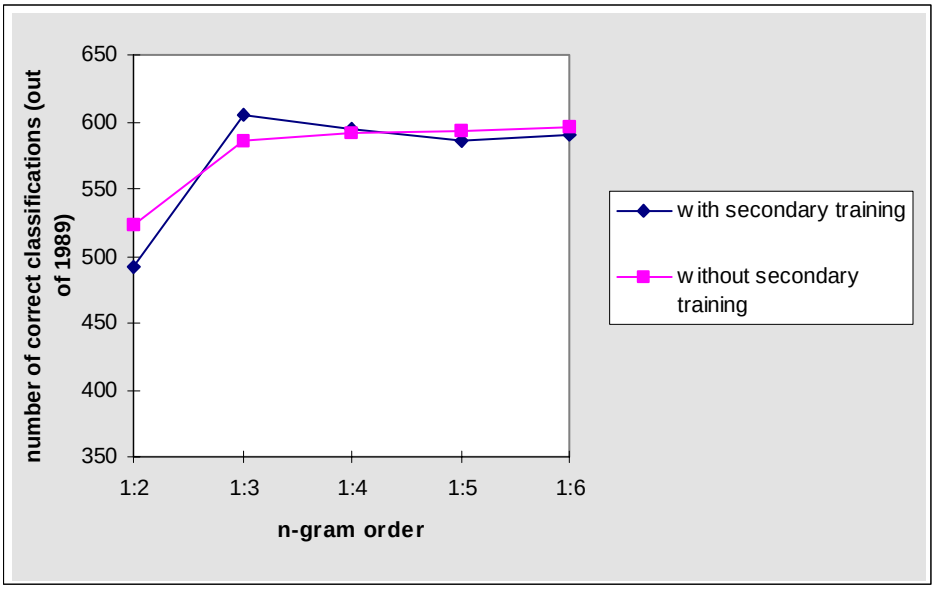

Figure 5.28 Classification performance with and without secondary training

Although there is not much divergence between the two sets of results, observe that the suppression of secondary training is only advantageous at a very low n-gram order where performance is poor in any case. At 1:3, which has by now been established as the n-gram order yielding the best performance for classification, secondary training makes a slightly more significant contribution.

5.7 Floor value experiments

When a sequence is found in one or more classes, but not every class, it is assumed that the probability of its occurrence in classes where it is not found is equal to the floor value. Every test utterance includes label sequences, at the higher n-gram orders, which are not found in every class in training data. Now, if the utterance contains one sequence not found in the *info* class, another not found in *prob*, and further sequences not found in each of the other four classes, it is only the application of the floor value that prevents an aggregate probability of zero being returned for all six classes in respect of that utterance.

The floor value was initially set to 10^{-6}: an arbitrary small number which is lower than the estimated probability of any prosodic label sequence associated with a particular class, at the n-

gram orders with which we are working. Better classification performance was, however, achieved with a much lower value such as 10^{-9}. Further decreases beyond that level had no effect, but in order to guarantee optimal results, the floor value was set to 10^{-12}. All experiments conducted thus far use the lower value.

Figure 5.29 displays the results for a comparative experiment (using both training sets and the 30 centroid model); it will be seen that the lower floor value performs better for all n-gram orders except 1:1 (where the floor value is never called upon, because all unigrams occur in the training data).

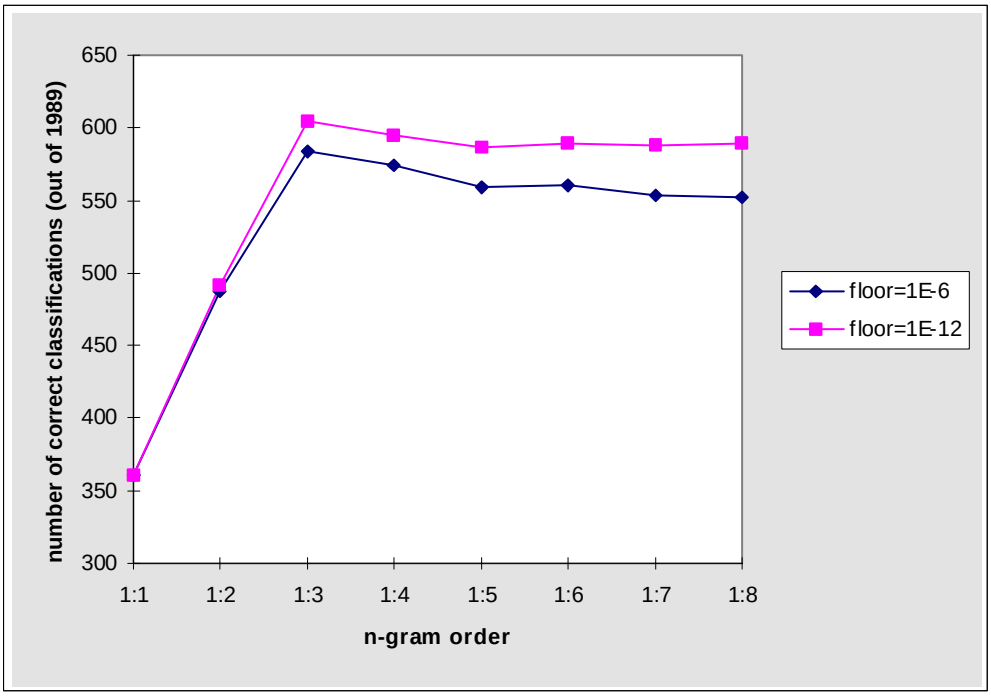

Figure 5.29 Comparison of classifications at two different floor values

The reason for this performance discrepancy lies in the aggregation of sequence probabilities in determining the most likely class for an entire utterance. The classifier algorithm steps through the utterance, maintaining an accumulator for each class, adding its contents to the class log probability of each new n-gram encountered. When the floor value is set to 10^{-6} (-6 in the log domain), the contents of the six class accumulators differ less widely from each other, with each increment, than when the value 10^{-12} (-12 in the log domain) is used. As a consequence, if, in test

utterance *u*, the accumulator for class *C* resorts occasionally to the floor value (because there are observed a number of n-grams that do not occur in *C*'s training data), it is more likely that *u* will be declared a member of *C* with 10^{-6} as the floor value than with 10^{-12}. Observe, from the intermediate program output in Table 5.4, that the utterance is most likely to belong to class 5 (*prob*) on the basis of the evidence so far; but that in the 10^{-6} case, it is going to be a great deal easier for some other class to usurp this lead.

Table 5.4 Intermediate algorithm step, showing accumulation of probabilities under two possible floor values, for the observed 5-gram sequence 17-2-7-4-10

	floor value = 10^{-12}		**floor value = 10^{-6}**	
	accumulated probability	*current probability*	*accumulated probability*	*current probability*
class 1	-20.604734	-12.000000	-14.604734	-6.000000
class 2	-28.906437	-12.000000	-16.906437	-6.000000
class 3	-28.967152	-12.000000	-16.967152	-6.000000
class 4	-21.231220	-12.000000	-15.231219	-6.000000
class 5	-13.405031	-4.480144	-13.405031	-4.480144
class 6	-20.378288	-12.000000	-14.378287	-6.000000

At the intermediate algorithm step shown in Table 5.4, the n-gram concerned occurs only in class 5. The accumulated probability applies *after* the current probability has been added in. On the LHS of the table, the class 5 accumulated probability is much greater than the contents of the competing classes' accumulators; it will be difficult for the competitors to recover from this, and

the utterance is, as it turns out, assigned to class 5. With the greater floor value, illustrated on the RHS, although class 5 again has the highest accumulated probability, other classes, particularly 1 and 6, are within reach. The utterance will eventually be assigned to class 1.

The foregoing establishes how different floor values put out different results. It has been shown, too, that the lower value yields a better classification performance. The following theoretical motivation for keeping the floor value as close to zero as possible is also offered.

In the first place, it does not strictly represent an estimate of the probability of the sequence absent from training data, since we have no real idea of what that probability might be. Rather, it is a device to allow us to make our calculations; without it, our efforts would be completely foiled by the intrusion of zero probabilities. Secondly, the probabilities of n-grams in training data sum to one over the class or classes in which they occur. The floor value, again, is not a true probability, as it is not included in this summation; by keeping it close to zero, its impact on the class probability mass is at least minimized.

5.8 Class prior weighting

The **class prior** refers to the likelihood that an utterance belongs to a particular class given the popularity of the class in training data. It is required because the six classes do not have equal numbers of utterances. Recall that none of the experiments reported thus far have made explicit use of the class prior.

However, it was shown in 5.5.3 above, where the random data experiments were reported, that rarer n-grams were less likely to occur in unpopular classes, in the training data; test utterances at higher n-gram orders invariably contain such n-grams. It follows, it was reasoned, that classification of observed utterances makes considerable appeal to the relative popularity of classes in the training data. Of course, this behaviour reflects the number of *n-grams* in each training class, rather than the number of utterances, as used to calculate the true prior; but one would scarcely expect the relative distributions of the two entities to differ very much.

It is expected, when the class prior proper is applied, that results for the unigram (1:1) models will improve quite dramatically, whilst the impact with higher order models will be less noticeable. This is because the surrogate class prior described above does not apply to unigrams, all of which occur in all training classes. This expectation is substantiated by results presented in Figure 5.31 below.

In the computation of class membership prediction, the posterior probability (that of the data given the class) is multiplied by the class prior. Because the estimate available for the posterior can be inaccurate if particular n-grams are poorly represented in the training data, the resulting classification can appear to depend far more on the class prior than the posterior probability. The effect of this is that practically all observed utterances are simply assigned to the most popular training data class. In order to allow the posterior and prior contributions to interact effectively, it was decided to weight the prior with a constant which yielded the best overall performance improvement. This was implemented by raising the prior to the power of the constant; in experiments reported so far, the constant was set to 0, so that the prior itself was levelled to 1 and ignored. A setting of 1 served to suppress the weighting effect.

A set of 200 experiments, based on the 30 centroid model, was conducted to determine the optimum value for the class prior weight, in which the classification performance at all parameter permutations of n-gram orders 1:1 to 1:5 was tested at a variety of weighting factors between 0 and 1. Moreover, each of these experiments was repeated with five different values for the minimum n-gram frequency parameter, which is discussed in the next section: 25 permutations were tested for each value of the prior weight. The value axis of Figure 5.30 shows the average of the 25 performances for each of 8 cases.

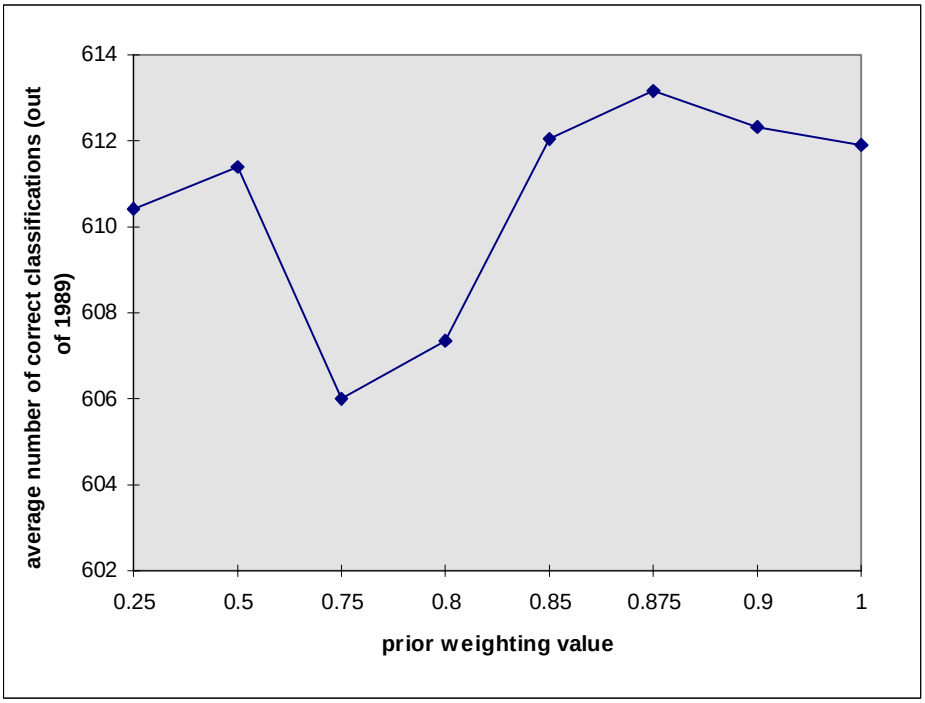

Figure 5.30 Aggregate results of a large number of experiments, in which different values for the class prior weight were compared

In a ninth case, the weight was set to 0 (that is, the prior was ignored). The average performance here was 521.48 correct classifications – as expected, a comparatively poor result.

Figure 5.31 includes the cases where the class prior was ignored, and presents comparative results for the different n-gram orders. Again, an average of all five values of the minimum n-gram frequency is computed for each data series. Notice how the 1:1 and 1:2 n-gram order curves climb sharply at the lower end of the category axis, demonstrating the importance of taking into account the class prior.

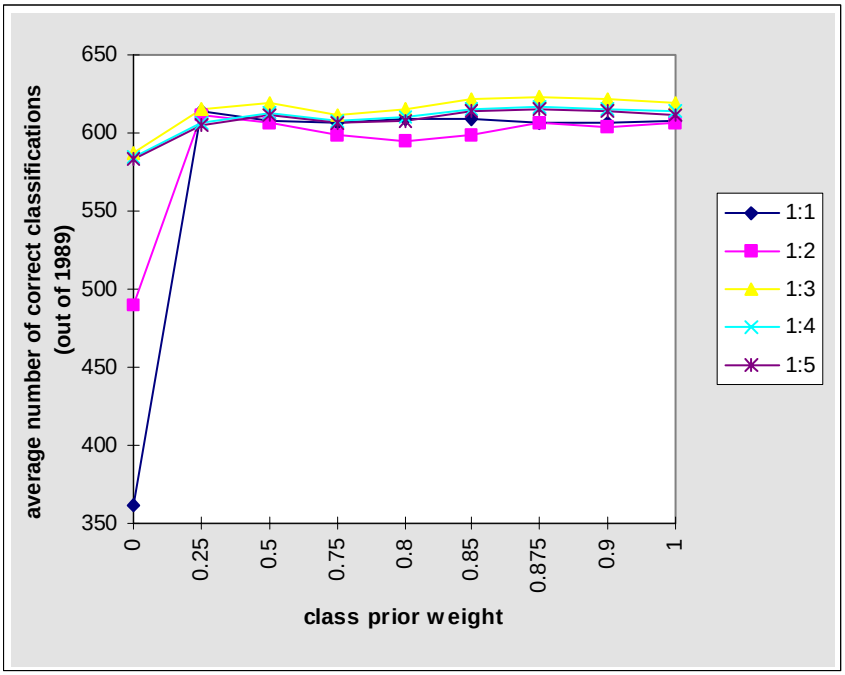

Figure 5.31 Class prior weight experiments presented by n-gram order

The conclusion that 0.875 is the most suitable prior weighting factor is reinforced by the fact that the single best performance of 645 (32.43%) correct classifications (presented in the next section) was at this level. This peak was reached using the n-gram order 1:3 and a minimum n-gram frequency of 2.

5.9 Data pruning

5.9.1 Minimum n-gram frequency

In the 1:3 n-gram order secondary training set, there are 92710 n-gram tokens: in other words, that is the total number of unigrams, bigrams and trigrams. This figure corresponds to a little under three times the number of segments (and of unigrams) in the training set. Of this total, only 4011 are hapax legomena (tokens which occur only once). Discarding these n-grams makes intuitive sense: they are so rare that they cannot be representing any true patterns in the data, and the class assignment in the training data is probably a matter of chance.

What is more surprising is that better results are not achieved by discarding even more n-grams.

There are 1753 n-gram types occurring twice, for example, giving 3506 tokens, and 949 types occurring 3 times (2847 tokens). The answer is that where all tokens of these n-gram types occur in just one class, there is much stronger evidence that an observation of such an n-gram characterizes that class; the more dispersed among the classes tokens of a given n-gram are, the less powerful a predictor the n-gram becomes. Of the 3506 tokens occurring twice, 1052, or just under a third, are only found in one class, while among tokens occurring three times, only 324 (of 2847) are unique to one class, a sharp drop.

Higher frequency n-grams, then, are important because they reflect the distribution of prosodic patterns among the classes. Hapax legomena are generally not useful because their appearance can be a matter of chance. Those n-grams occurring twice have substantial predictive power, especially, perhaps, where both tokens appear in one class.

Figure 5.32 plots the n-gram order and minimum frequency data, using a class prior weight of 0.875. At 1:1, the performance is the same across the board, because there is no n-gram at this order that occurs less than 5 times (there are only 30 possible unigrams). At 1:2 there are still only 34 hapax, so again the curve is rather flat. At 1:3, performance peaks for the minimum frequency of 2; n-gram orders 1:4 and 1:5 also perform quite well here, dip at 4, and then climb. Since a large proportion of four- and five-grams are hapax, one might expect only decline at minimum frequencies of 2 or more. But the sheer abundance of higher order n-grams means that *some* of them do occur more than once. Where that happens, it is very likely that some prosodic pattern which is representative of a particular class has been discerned.

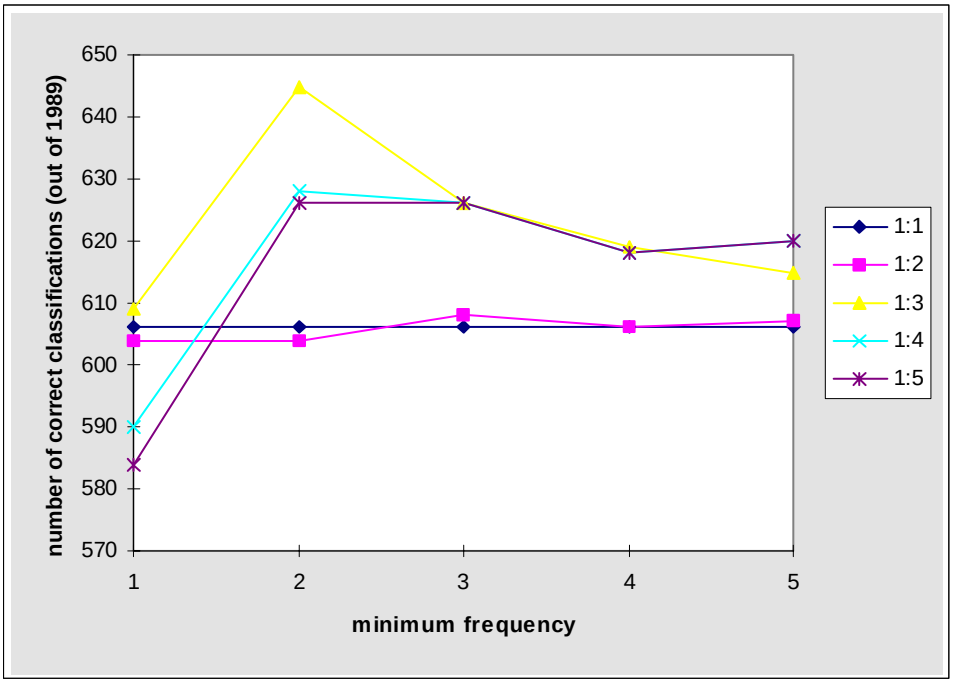

Figure 5.32 Performance under different n-gram orders and minimum frequencies

5.9.2 Salience thresholding

Every n-gram in the training data is assigned a salience score, as specified in Chapter 4, which is the maximum probability of each class given that particular n-gram. The salience, therefore, gives an indication not of the n-gram's degree of association with a particular class, but of its discriminative power for the classification task generally. Experiments were conducted with different values for the salience threshold, chosen in such a way that an equal proportion of the data was admitted with each increment of the parameter; thus, given the fifteen non-zero settings used, a threshold of 0 treats all the data, 0.008 treats all but the least salient fifteenth of the data, and so on.

Figure 5.33 shows the results of these experiments. The reason that the performance is worse than the best result reported so far is that the value of *absmin* is restored to zero.

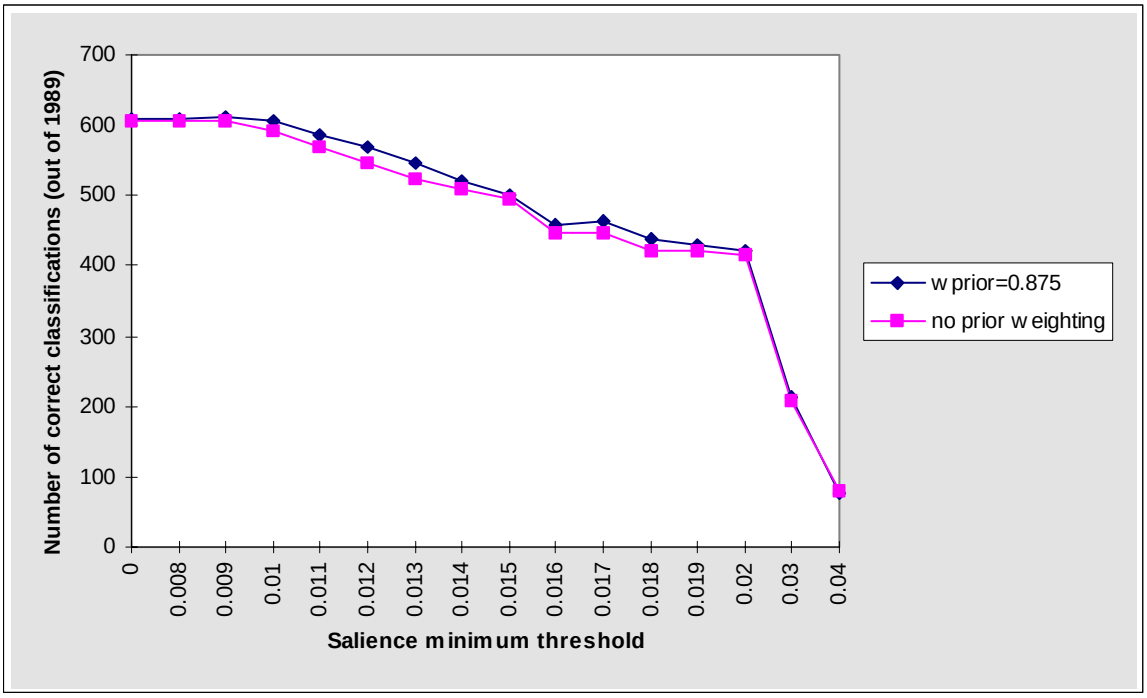

Figure 5.33 Results of salience threshold varying experiments, for sixteen different thresholds (including 0, no thresholding), with and without class prior weighting

As one would expect, performance is uniformly better with class prior weighting than without. The best performance is a threshold level of 0.009, with 611 (30.72%) correct classifications; this however, is hardly significantly better than the 609 (30.62%) correct classifications achieved without any thresholding. The overall trend, clearly, is for performance to deteriorate with each parameter increment, so that when the threshold is set to a very high value, it becomes very poor (note that the apparently steep performance decline at the two highest settings is due to the non-linear scale of the plot).

The difficulty is that even though the n-grams excluded are non-salient, they may be relatively common: the least salient n-gram in the 1:3 data is the bigram 01-21, which occurs 28 times. The unigrams, especially, are very non-salient: half of them are in the least salient fifth of the data. The effect of this is that many of the shorter utterances in the corpus contain only non-salient n-grams, and cannot be classified at any but the very lowest salience thresholds. One solution to this problem would have been to automatically assign such utterances to *info*, the most popular class;

this would have boosted performance, but at the expense of methodological propriety.

The experiments described in this section did not exploit the *absmin* parameter – if an n-gram occurred only once in the training data, it was taken into account. It is obvious that adjusting this parameter upwards would only increase the number of unclassifiable utterances, so no such experiment was performed.

The failure of the salience experiments is essentially attributable to a data sparseness problem. Note that this was also the finding of the topic identification experiments of Chapter 2, with respect to salience and other data pruning techniques.

5.9.3 Mutual information thresholding

Since MI is a data pruning technique that does not appeal to absolute frequency of occurrence of tokens, like salience, optimism about its contribution to classification performance is probably not warranted. Experimental results are nevertheless included for the sake of completeness.

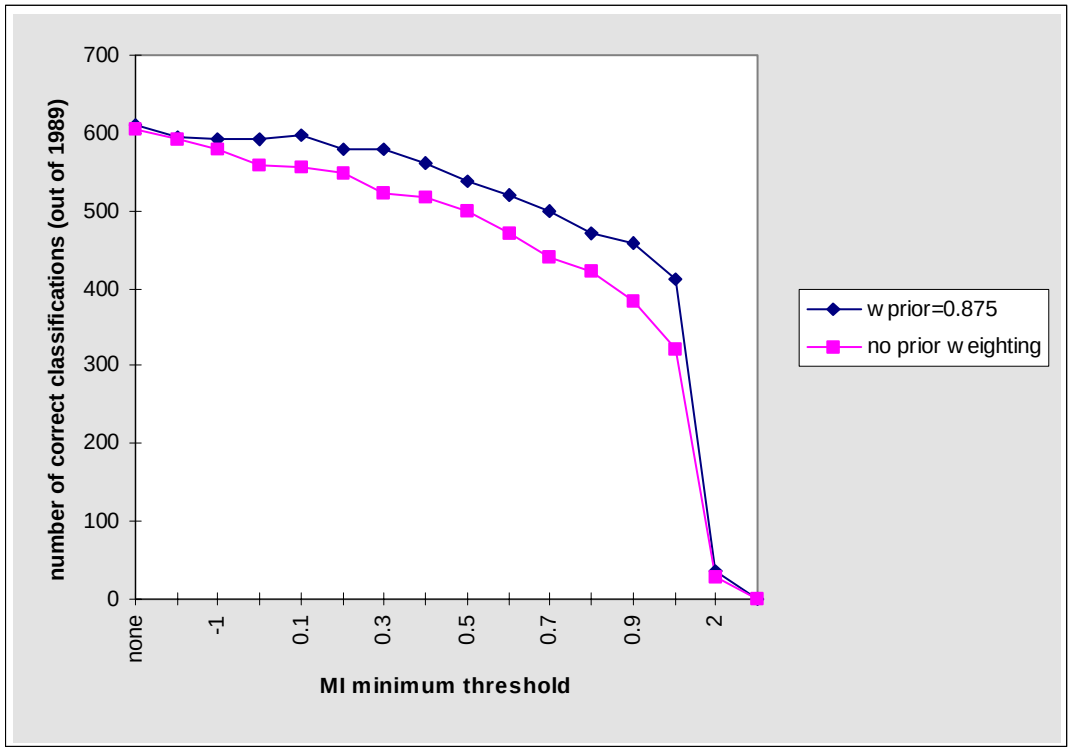

Figure 5.34 Results of MI threshold varying experiments, for sixteen different thresholds (including -∝, no thresholding), with and without class prior weighting

Again, notice the improved performance with the class prior weighting. One reason for the deterioration in performance with each parameter increment is that a small number of utterances (60 out of 1989) were so short that they contained only one segment. MI is a measure of the degree of collocation of the final prosodic label of a sequence and the labels preceding it: where an utterance bears only one label, the measure becomes vacuous and the utterance cannot be classified. As noted, the data sparseness problem encountered with salience applies here also.

5.10 Summary

This chapter has presented the results of a fairly large number of experiments, which consisted mainly of the adjustment of PLoNQ system parameters. Two groups of these were designated: training parameters, generally optimized by the inspection and analysis of intermediate system output, and test parameters, whose ideal values were determined in the main by their ability to maximize the number of correct classifications of unseen utterances. By necessity, some of the analysis of training parameter performance was intuitive and possibly subjective.

Only a subset of a potentially huge number of experiments was conducted; an exhaustive approach would have entailed a combinatorial explosion in the number of experiments.

The final, optimal, combination of parameter settings was given in Table 1. By this combination, 645 unseen utterances out of 1989 (32.43%) were assigned to the correct utterance type class. While this performance is by no means spectacular, it constitutes a considerable improvement over chance (30.02%), especially when it is borne in mind that only prosody, and no other linguistic stratum, was taken into account in the experimentation.

In the next chapter, we turn to an alternative experimental approach, using human subjects.

6. Testing psychological reality: a psycholinguistic experiment

6.1 Background to the experiment

It has been demonstrated that the PLoNQ system has an ability (albeit a limited one) to classify utterances on the basis of prosodic information. An important question emerges from this finding: are we asking too much of the computer? Is the information needed to determine UT simply absent from the signal? One way to find out is to ask human informants to perform broadly the same task, and see how the two sets of results compare.

Of course, computers and people enter the fray on somewhat unequal terms. The computer reviews a mass of domain-specific data each time it is called upon to make a decision, a large enough quantity to completely overwhelm an informant. Still, the informant does bring to the task their own personal ontology, a vast resource consisting of the knowledge, experiences and intuitions of a lifetime. The resource incorporates native-speaker control of at least one language, and a profound knowledge of its prosodic structure.

The participants would be almost certain, with the best will in the world, to take into account lexical information (perhaps to the exclusion of all else) in arriving at their classification judgements. So, it would be necessary to play them a version of the utterances where all but prosodic information was suppressed from the acoustic signal (a fourth order band-pass filter at 30 to 300 Hz is what was in fact used; this software is part of the SFS package). This solution is more than a little artificial, for speakers are not used to this type of signal, relying normally on prosodic effects solely as an adjunct to segmental information; Abercrombie (1967) does report on some tone languages in which meaning can be conveyed by whistling and drumming, but we are confident that none of our participants had been exposed to such exotica. Nonetheless, it is so widely accepted that prosody carries attitudinal and illocutionary information, it seemed plausible that participants would be able to tease the information out.

Prosody research has been the focus of much psycholinguistic experimentation. Some of those working on stylization of F_0 curves, including Campione & Véronis (2001), were able to re-synthesize the resulting contour and apply it to the original acoustic signal. Participants could be asked whether they could detect a difference between the stylized signal and the original. Campione & Véronis used the MOMEL model of stylization, which is reported in detail in the next chapter. Pijper's work (1983) is an account of various perceptual studies based on re-synthesis.

Ladd et al (1986) used masking techniques similar to ours, in order to elicit the attitudinal flavour of a speech signal. They recorded social security interviews (using welfare staff, but actors stood as clients). Participants were exposed to a written transcription of the exchanges, a normal recording, and recordings that had been subjected to three types of masking: low-pass filtering, random splicing and playing backwards. Of these, only the first preserves the F_0 curve. They found a negative correlation between attitudes perceived in the transcription and normal speech, and a positive correlation between normal speech and speech which had been masked, confirming an association between prosodic and affective features.

Levin et al (1982) applied a low-pass filter to recordings of read speech and spontaneous story-telling. They found that listeners could readily tell the difference between the two speaking styles, even without segmental information.

6.2 Experimental method

Sixteen participants, two of whom were women, took part in the experiments, which were conducted individually, in an office environment with limited background noise. All were native speakers of British English, and all were research students or staff within this School. A small proportion of the participants were themselves involved with speech research, but not prosodic analysis. They were not told the purpose of the experiment until they had completed all the tasks; partly to prevent them from over-zealously seeking out prosodic patterns, and secondarily because it was of interest to see whether they could deduce what linguistic information they were being

asked to rely on.

The participants were presented with a simple point and click interface, specially written for the task, [3] and listened to the audio data over headphones. They were not required to speak, write or type. There were five parts to the experiment, which took about 40 minutes in total. Before starting, Figure 6.1 was presented on-screen to the participants, to explain the procedure.

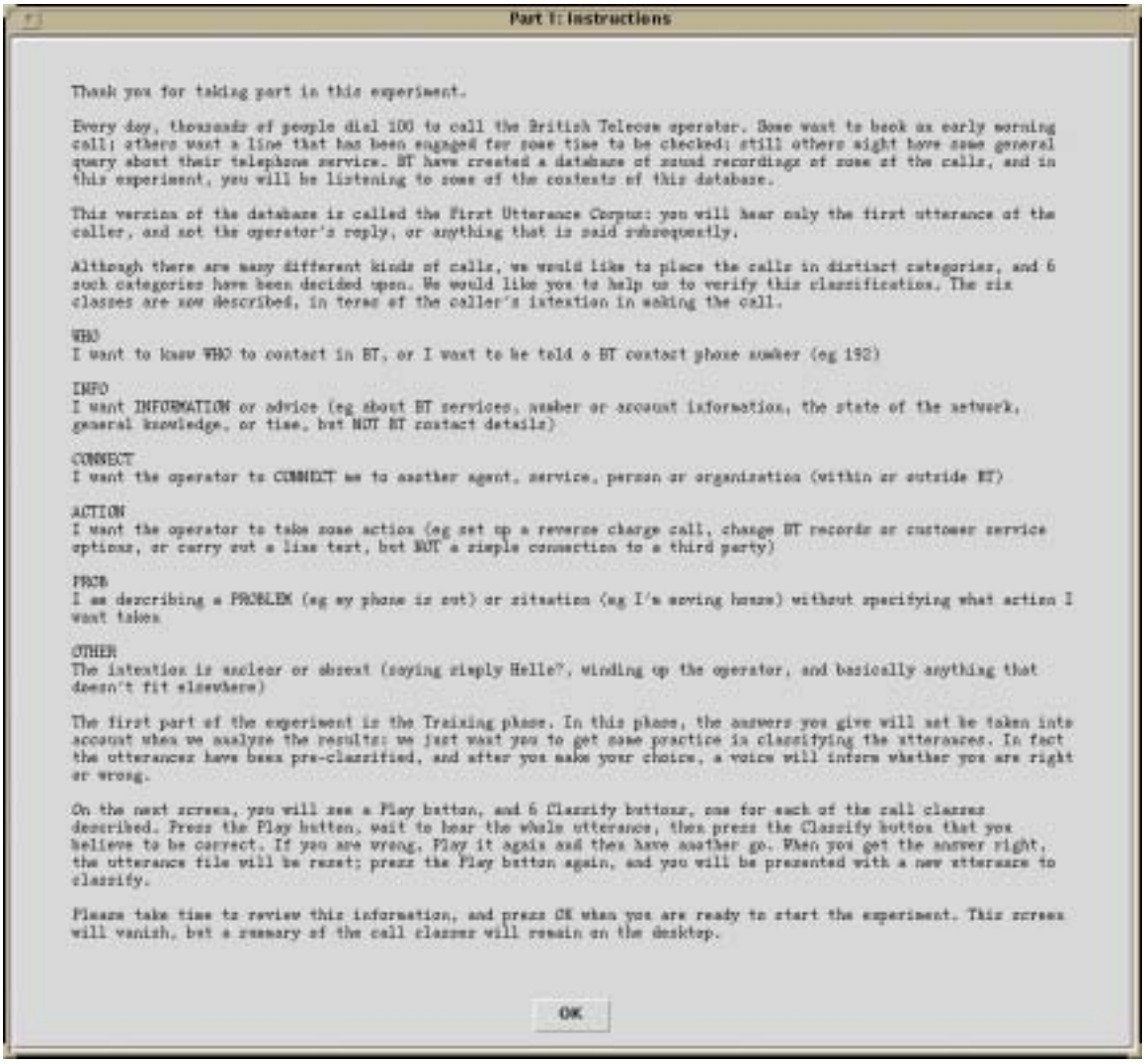

Figure 6.1 Instructions explaining experimental procedure to participants

[3] The Tcl/TK script used was developed from a template kindly supplied by Jin Jian-hong.

6.2.1 Training session

The first part was a training session, where the participants were played 25 utterances from the Oasis corpus, without filtering, and asked to classify them according to the six Oasis UTs. They were able to replay the utterances as often as they wished, but could not continue with a new utterance until the active one had been correctly classified, according to the BT annotation. This gave participants the necessary understanding of the annotation scheme. Figure 6.2 shows the point and click interface used for classification, and Figure 6.3 the UT prompt bar that remained on the desktop for the duration of the UT classification task.

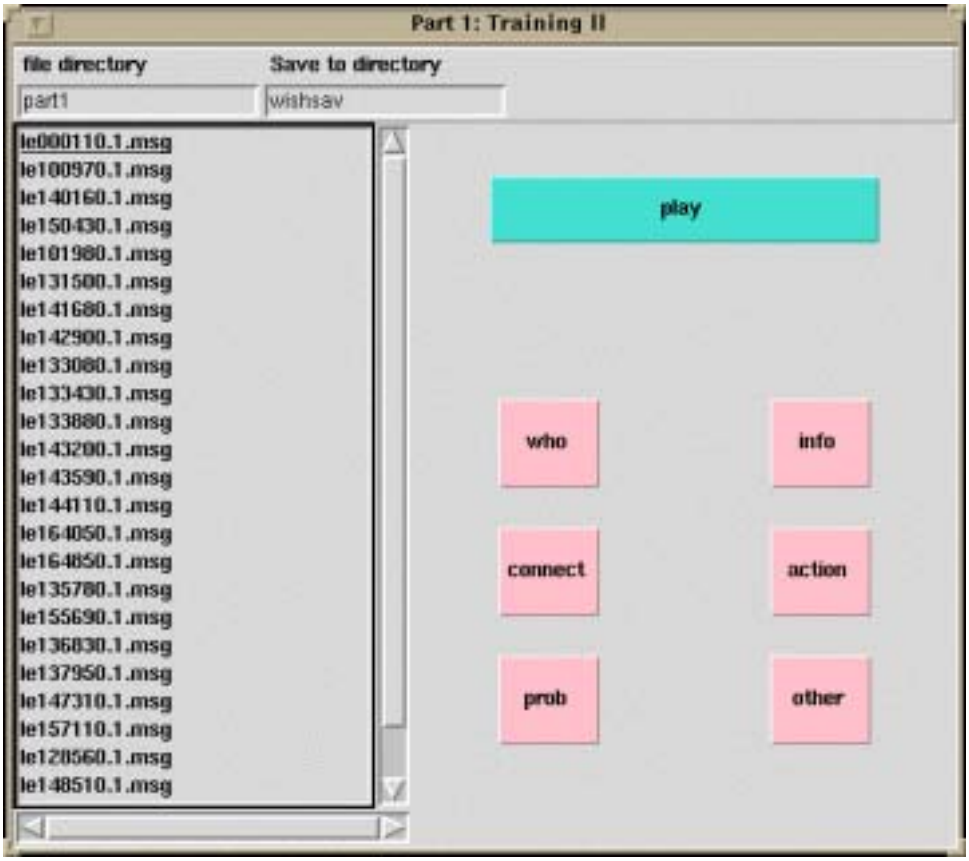

Figure 6.2 Point and click interface for experiment

Figure 6.3 Desktop UT prompt for participants

6.2.2 Classification of unfiltered utterances

Participants were asked to classify 25 further normally recorded utterances, and this time their scores were noted; all participants were played the same utterances, but the order in which they were presented was varied. This part of the experiment served three purposes. First, it provided reinforcement of the earlier training phase. Secondly, it enabled the experimenter to weight the scores of the filtered utterances of the next phase appropriately: performance on that phase needs to reflect the fact that the scores are not perfect even when the utterances are played normally.

Finally, it provided a useful gauge of the psychological reality of the BT annotation. Error patterns in the BT annotation were, it turned out, reflected in the classification efforts of the participants. Some objections were voiced, too, about the intuitiveness of the Oasis scheme.

Among the 25 utterances were two that had been incorrectly annotated in the corpus. They are presented in Corpus Excerpt 6.12 and Corpus Excerpt 6.13.

phase2/day02/le150440,B,,,phase2/day02/le150440.1.msg,,who,0,~ @@ could you put me on to someone to help me with an address love please @@,other-op,,,,,5,,,2,2.077,5.224,poor,good

Corpus Excerpt 6.12

In Corpus Excerpt 6.12, the caller is requesting connection. The confusion probably arises because the caller believes someone within BT can help, and *who* is the correct category for requesting BT contact details. Nine participants selected the correct *connect* class, with five people making the same error as the annotator.

phase2/day03/le166270,B,,,phase2/day03/le166270.1.msg,,prob,0,#! <cough> oh operator # um ~ @@ i have a number here (mm) and it's only five digits # (right) and it was given to me in 94 so what would have changed @@,dq-area-old,,,,,5,,,2,1.874,10.015,ok,good

Corpus Excerpt 6.13

In Corpus Excerpt 6.13, we have a request for information. This time, all but two of the participants made the correct classification.

Another situation that causes some confusion is where a caller wants the operator to help her with a third party number, as noted in Chapter 3. The operator will conduct a line check (calling a number to see if there is conversation) in response to an *info* request ("Can you tell me if there's someone talking...") or an *action* request ("Please will you check..."). The *connect* class can be used for a request to put the caller through to a third party, which is essentially the same task, given that she will only request this action when she has been experiencing difficulty dialling the call herself. In fact, a semantically identical request could equally be cast in the *prob* category: "I'm having problems getting through to 01..."

Annotators – or informants – have the following dilemma. On the one hand, they are encouraged not to interpret what the caller says too literally, such that "Can I make a reverse charge call please?" should not be classed as *info*, for example. Then again, they are being asked to analyse the words used by the caller, and not perform the classification on the basis of intentions alone. The line check call ambivalence demonstrates this amply.

In general, the participants enjoyed the task, while finding it reasonably straightforward.

6.2.3 Classification of filtered utterances

Participants found this part of the task a good deal more challenging. The experiment was conducted in exactly the same way as the last; a different set of utterances was used, however. Figure 6.4 shows the procedure as presented to participants, for the reader's reference.

Figure 6.4 Instructions for classification of filtered utterances

The volume control referred to was also on the desktop. Although the order of play was randomized, all participants found that the task gradually became more tractable as they proceeded.

The participants reported various strategies for making their choices. Although the sound was extremely muffled, the occasional word could in fact be made out: in one of the utterances, for example, *please* could be discerned by some participants at the end, and this led them to (correctly) classify the call as *action*. Duration was seen as an important clue, with many of the participants preferring *info* or *prob* for longer queries, *action* or *other* for shorter ones.

The most important factor, according to all participants, was a phenomenon described variously as "tone of voice", "sound frequency", "the way the voice goes up and down", "intonation" and in one case "prosodic effects". This, clearly, is the response we hoped to elicit. In particular, participants St and Pa used a rising intonational contour to be indicative of a question, which in turn they thought was most likely to instantiate the *info* UT. At the affective level, Li and Mi discerned greater and lesser degrees of assertiveness from the prosodic features, and drew upon their experience with the unfiltered task to classify what they perceived as more hesitantly enunciated queries as *prob*, and to choose *action*, for example, when they detected confidence. The reasoning behind this strategy was that the confident caller, speaking assertively, has a clear idea what they want the operator to do, while the less assertive *prob* caller is making a more

tentative approach to the operator, unsure of what action they want taken.

It is a pity that audio examples cannot be included in the thesis[4]: the reader would surely agree that, while the unfiltered utterances are entirely incomprehensible in terms of lexical content, prosodic patterns which could be labelled (for example) "querulous", "hesitant", "confident" and "bored" do indeed emerge.

6.2.4 Classification of isolated words

A possible criticism of the experimental work presented so far in this chapter is that no empirical evidence of the low-pass filter's effectiveness has been presented. In the final two parts of the experiment, therefore, participants were asked to identify individual words. In Part 4 the words were filtered, and in Part 5 they were not. Multiple tokens of four words were extracted from the corpus.

One way of conducting this part of the experiment would have been to ask the participants to type in the word they heard, without any restriction as to what that word might be. This, though, would have differed too much in nature from the previous classification tasks, as well as introducing the possibility of typographical errors, so it was decided to use words from a pre-specified set; after hearing the word, the participant simply clicked the appropriate button to indicate their choice. In order to make the task a little more challenging, pairs of words that were phonetically similar were chosen (*phonebook* and *number*, *morning* and *ringing*). That the words are all of the same number of syllables needs no explanation.

In principle, it is easier to identify word types from a closed set than utterance types. An analogous task, with whole utterances, would have been to ask the participant to listen and then select the corresponding sentence from a set displayed on the screen. This was not attempted, but it is surmised that participants would have attained near perfect performance in such a task.

---

[4] For legal reasons.

The central hypothesis of this part of the experiment was that performance on the filtered words would be worse than on the filtered utterance classification. The unfiltered word set constituted a benchmark only, and perfect results were expected: in the event, one participant identified one of these words incorrectly. The participant himself ascribed this error to inattention.

Participants felt that, given the two filtered tasks, it was indeed more difficult to identify words than classify utterances, although participant Ti claimed that he used prosodic cues in the word task too: *morning*, apparently, had a rising tone, while *ringing* did not.

6.3 Results

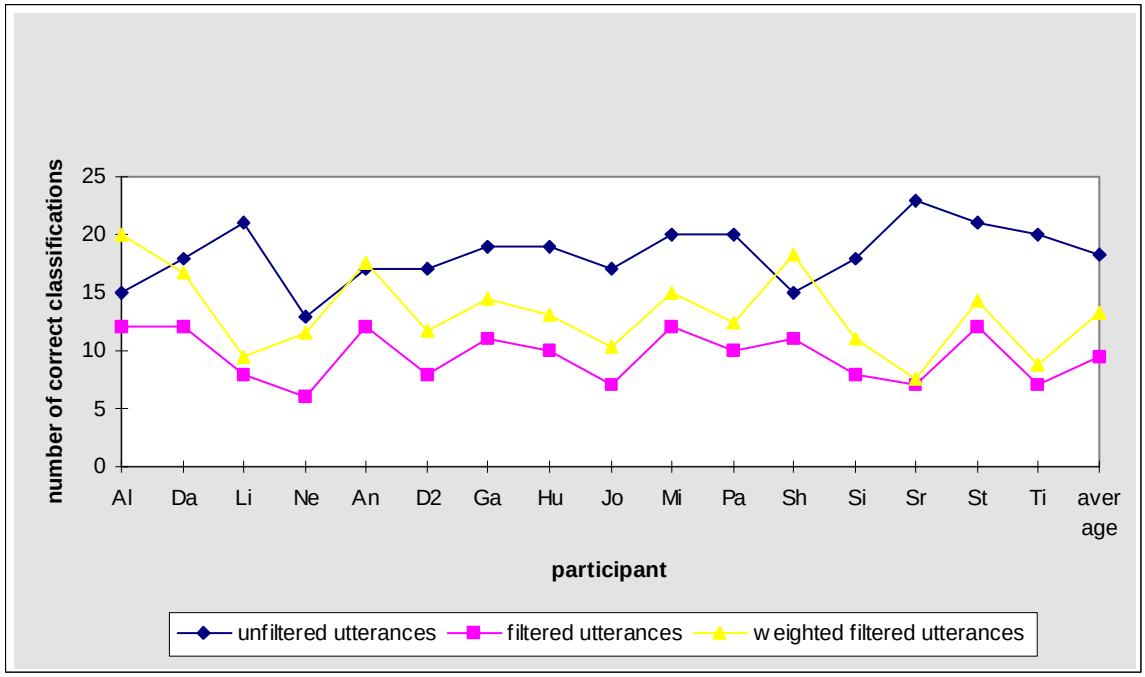

Figure 6.5 Number of correct utterance classifications, by participant (out of a maximum possible score of 25)

Figure 6.5 shows that all participants performed better at classifying unfiltered than filtered utterances. This comes as no surprise, of course. However, since even the unfiltered classification was not perfect, it was decided to weight the filtered performance of each participant with an index of their competence at the unfiltered task. Note, now, that the weighted filtered performance

of three individuals (Al, An and Sh) exceeds their performance with the normal utterances. The weighted performance gives us an indication of how well each individual exploited the prosodic signal in the speech stream, normalizing by their understanding of the task. The weighted scores were calculated by simply dividing the filtered performance score by the score on the unfiltered task, and multiplying by 25 (the number of utterances in the tasks).

The average performances were: unfiltered 18.31 (73.24%); filtered 9.56 (38.24%); weighted filtered 13.29 (53.16%).

In the last section, it was hypothesized that participants should find it markedly more difficult to identify words, because of the probable absence of prosodic information in the single word case. Whether this expectation was confirmed depends on how one analyses the results, which are shown in Figure 6.6.

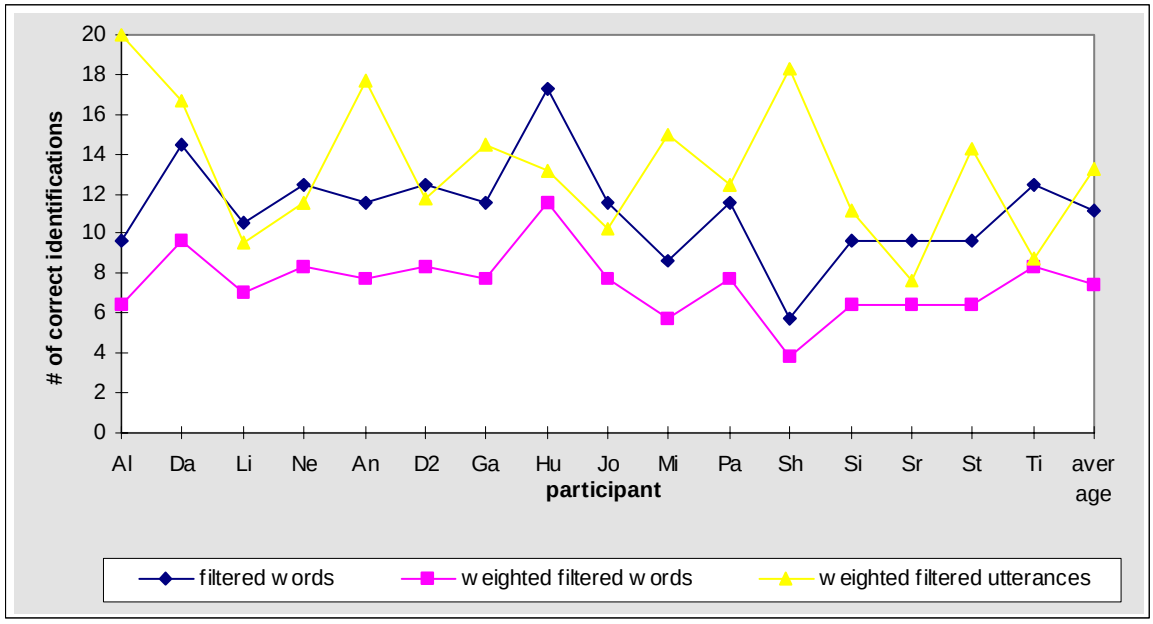

Figure 6.6 Number of correct classifications, by participant (out of a maximum score of 25)

The filtered words are more difficult to identify than filtered utterances are to classify for nine of the sixteen participants, and the average success rate for filtered words is 11.18, compared to 13.29, as noted above, for the utterances (when weighting is applied). Participant Sh seemed to find the word task especially challenging compared to her performance on utterance classification.

When the filtered word scores are weighted, performance on this task is substantially worse in all cases, with an average success rate of only 7.45 (29.8%). The weighting procedure is different in this case than that employed for utterances: there is no need to take into account performance on the related unfiltered task, as perfect scores were attained there in virtually every case. However, the fact that only four word types were offered, as against six utterance types, needs to be taken into account; the weighting recognizes this, and simply multiplies the unweighted scores by two thirds.

6.4 Summary

The experiment provided useful feedback on the Oasis annotation. In the unfiltered classification task, errors made by the participants mirrored the patterns of occasional annotation errors in the corpus itself. The main purpose of the experiment, though, was to determine the degree of psychological reality of the PLoNQ prosody-based utterance classification.

The best performance obtained by the PLoNQ system, it will be recalled from Chapter 5, was 645 correct classifications out of 1989 (32.43%), with chance at 597 (30.01%). In this experiment, the weighted average performance of participants was 13.29 (53.16%) correct classifications; clearly a substantial improvement over automatic performance. If the weighting is disregarded, the average result, 9.56, corresponds to 38.25%. There were 9 instances of the *info* class in the task, so successful classification by chance is slightly below actual performance at 9.0 (36%). These results do appear to confirm the psychological reality of the computational experiments.

7. MOMEL and INTSINT: an alternative implementation

This chapter presents comparative experiments using an alternative system of prosodic labelling to the linear trajectory and K-means clustering approach described in the foregoing chapters. In these experiments, MOMEL (MOdélisation MELodique) is used to generate a stylized contour, and the INTSINT (INternational Transcription System for INTonation) discretization scheme is used to generate the prosodic labels. Both schemes are documented in detail in Hirst et al (2000) and Campione (2001), and the account of the system give here is essentially paraphrased from those two sources. They were developed by Daniel Hirst and colleagues at Aix-en-Provence.

7.1 MOMEL stylization

A stylized contour is in a sense a target F_0 curve: it should represent the pitch modulation pattern that the speaker is aiming at for the utterance as a whole, factoring out, as Hirst et al put it, micro-prosodic effects, which are characteristics of particular phonetic units. They claim that vowels and sonorants (a group of consonants which includes nasals and laterals) exhibit fewer micro-prosodic effects than other phones, and that consequently an utterance such as “Molly may marry Larry” would be given a stylization very similar to the original F_0 curve. Such a curve is continuous and smooth, and Hirst et al argue that a parabolic stylization constitutes a more accurate representation of it than a linear model, such as that presented in the foregoing chapters of this thesis. What is more, they claim, such a representation is more parsimonious, as they show in an illustration reproduced as Figure 7.1.

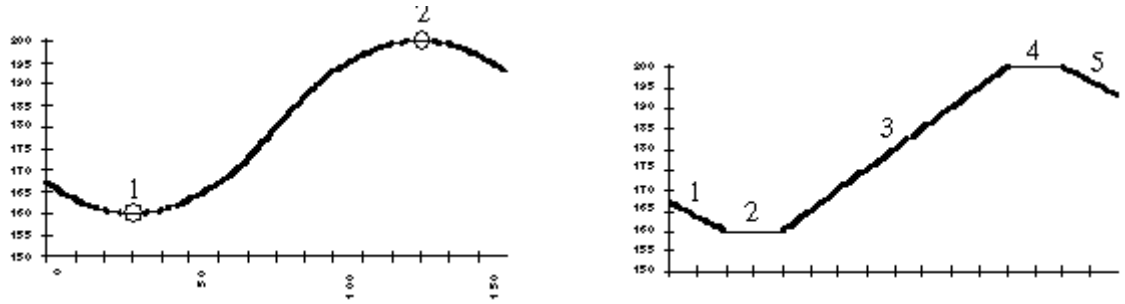

Figure 7.1 Hirst et al's comparison of linear (right) and parabolic stylizations

The same prosodic information can be represented by two **target points** (TPs) on a parabola, as opposed to the five segments required in the linear case.

The new contour can be used to re-synthesize speech, and the listener should not be able to tell the difference (although ’t Hart (1991) asserts that listeners cannot tell the difference between this kind of re-synthesis and one based on linear interpolation). Thus, MOMEL has a degree of phonological reality that a purely data-driven system (such as the early modules of PLoNQ) lacks. It exploits knowledge about the kind of segments that are subject to micro-prosodic effects, and what those effects might be; furthermore, it attempts a close copy of speaker intentions, while our system is geared to maximizing classification performance.

Hirst et al characterize the task performed by INTSINT as *phonetic* prosodic transcription, unlike ToBI or Tilt which are *phonemic* in nature. Different versions of ToBI, for example, have been trained and prepared for varieties of English and other languages, whereas INTSINT is shown to have multilingual coverage potential. The analogy here, of course, is with phonemic and phonetic transcriptions at the segmental level.

As re-synthesis is not a goal of our research, the precise method for fitting the curvilinear contour to the target points, a quadratic spline function, is not of immediate concern to us. The location of

the target points is important, as this is the information that is passed to INTSINT for discretization, that is the assignment of prosodic labels. Once labelled, the data is treated exactly the same as the output from K-means clustering in our own system architecture; it is passed to the PLoNQ n-gram module.

7.1.1 Computation of MOMEL TPs

The MOMEL stylization consists of four stages, which are described in the following sections.

7.1.1.1 Preprocessing of F_0

In order to remove spurious values at voice onset, any F_0 readings that are more than 5% higher than both their neighbours are levelled to zero.

7.1.1.2 Estimation of TP candidates

The following parameters are used in this stage: Δ (5%), *estWindowSize* (300ms), *partWindowSize* (200ms), *hzMin* (set to 50Hz) and *hzMax* (an adaptive threshold, explained shortly).

A window of length *estWindowSize*, centred on each F_0 frame in turn, is moved across the utterance. *hzMax* is computed as the mean of the top 5% of all readings within the window multiplied by 1.3. Any reading which is outside the *hzMin* to *hzMax* range is neutralized (that is, treated as missing).

The next step, still over the active window, is a form of quadratic regression, which Hirst et al term **modal regression**, because it stands in the same relationship to arithmetic mode as normal regression does to arithmetic mean. Rather than minimize the sum of squared distances, the modal regression function finds the series of values that are closer than Δ to the observed F_0 readings in the maximum number of cases. Hirst et al further require that there be no F_0 readings more than Δ above the regression contour; this is because they consider that micro-prosodic effects lower, and do not raise, the underlying macro-prosodic contour. Thus, the closeness constraint actually

applies only to readings below the contour.

Those observed F_0 readings that exceed Δ are neutralized, and the regression and neutralization steps are re-executed until no further neutralization is possible. The TP for the regression contour in the active window is then computed.

The steps of this subsection are repeated for each window, so that one candidate TP, a vector of time and F_0, is associated with every F_0 observation.

7.1.1.3 Partitioning

The purpose of this stage is to segment the sequence x of candidate TPs $x_1,\ldots,x_T$. A window of length *partWindowSize* is moved across the sequence, centring on each x in turn. Within each window, the averages of the candidate TPs on the left and on the right sides of the window are computed, and the *difference* between the two averages is associated with x. Then, x is a segment boundary if *difference*(x) is greater than the *difference* associated with both of its neighbours in the sequence, and greater than the average of all *difference*(x) in the utterance.

7.1.1.4 Reduction of candidate TPs

This stage entails the removal of outlier TPs, defined as those whose F_0 or time value differs from the mean of all TPs in the segment by more than one standard deviation. The mean of the remaining TPs is taken to be the final estimate of F_0 and time for that segment, and a further quadratic spline computation plots the MOMEL contour.

7.2 Assignment of INTSINT labels

The INTSINT algorithm is described in Hirst (personal communication), as well as the two sources named above. Again, most of what follows in this section is taken from these sources.

One way in which the algorithm attempts to optimize the prosodic labelling is by modelling the pitch *key* and *range* of the speaker. These parameters are obtained by exhaustively searching a parameter space 50Hz above and below the speaker's mean F_0, as well as a dynamic range between 0.5 and 2.5 octaves (the INTSINT program converts F_0 readings to the logarithmic scale). Of these parameter settings, those finally selected are those which minimize the error between the posited labelling and the TP values previously determined by MOMEL.

The symbolic names assigned by INTSINT are as follows: T (top); M (mid); B (bottom); H (higher); S (same); L (lower); U (upstepped); and D (downstepped). Of these, the first three are characterized as **absolute**, the others as **relative**. Initially, the predicted value of M is set to the *key* parameter.

The first TP, and those that are more than 500ms after the preceding TP are first considered (in the latter case, it is assumed that these TPs follow a pause – this assumption has to be made because no actual measurement of pause duration is taken, as the algorithm's input is confined to the F_0 curve). The predicted value of M is set to that of the *key* parameter, T is set to *key* + *range*/2, and B to *key* - *range*/2. For other TPs, the predicted value of M, T and B is set as for initial TPs, for H it is set to P + (T-P)/2, where P is the value of the preceding TP. This has the effect of placing H halfway between P and T.

The predicted value of L is P - (P-B)/2, of U, P + (T-P)/4 and of D, P - (P-B)/4. Finally, the predicted value of S is set to P.

In each case, the tone is selected which has the predicted value closest to the observed value. The squared error between the two is accumulated with the total error for the parameter values under consideration, and the final labelling is that which minimizes this error.

7.2.1 Algorithm parameters

Within INTSINT, the speaker's *range* and *key* are user-specifiable parameters, as is the size of steps through these parameter spaces for each iteration. The minimum distance between TPs for a

pause to be inferred may also be specified, along with the denominators of the relative tones H, L, U and D. However, as the parameter settings noted in the previous section were found by Hirst et al to be optimal, we have implemented these default values in our experiments.

In the linear trajectory phase of PLoNQ, key parameters control the length of segments to which labels are to be assigned. In the clustering phase, the number of centroids (in effect the number of prosodic labels available) is also specified by the user. With the MOMEL and INTSINT implementation, these values are not variables; rather, they are integral to the algorithm itself. The segmentation is largely at the syllable level, because of the rule-based nature of MOMEL, while the phonetic motivation of INTSINT dictates a discrete set of label types, eight in number.

Another significant difference between the PLoNQ and Hirst treatments is that in the latter there is no opportunity or need for distinct primary and secondary training phases. In PLoNQ, the goal of primary training was to establish, in an unsupervised fashion, the parameters associated with prosodic labels whose only predetermined characteristic was their number. In the Hirst treatment, the labels are applied to the data by rule.

7.3 MOMEL and INTSINT output

Figure 7.2 shows output for the five syllable utterance “Yes, Manchester please.” In this figure, the first pane shows the speech waveform and F_0 curve; it will be observed that all but the last TP, shown in the third pane, correspond to a syllable boundary. The MOMEL contour itself is shown in the fourth pane.

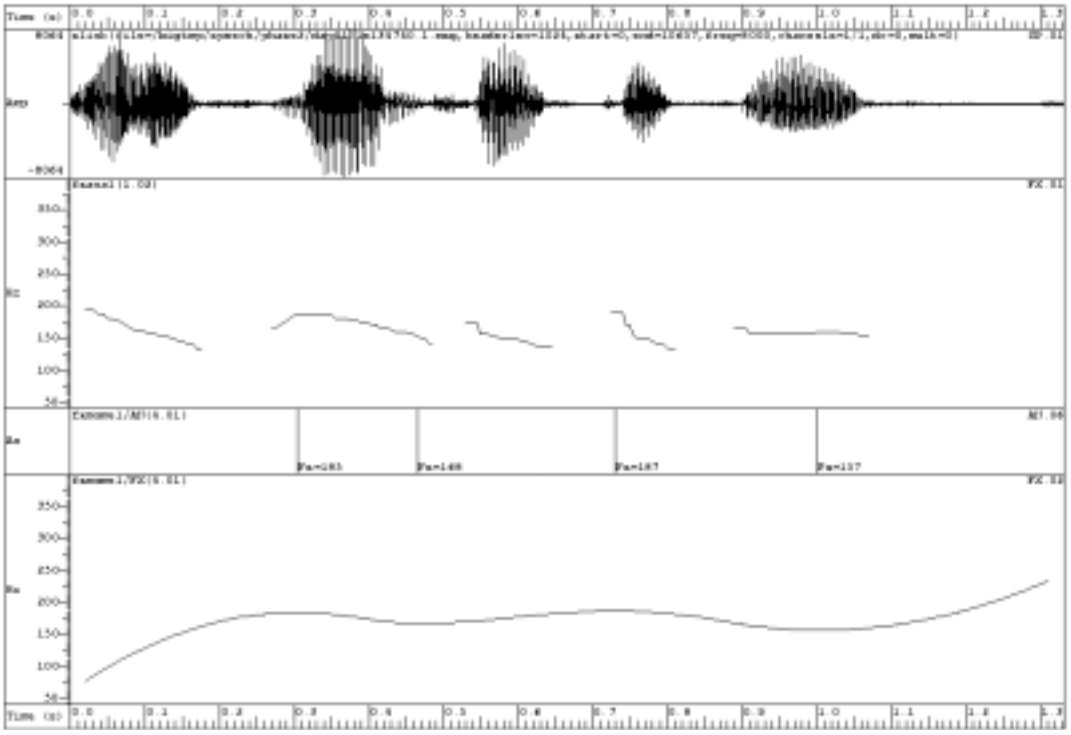

Figure 7.2 "Yes, Manchester please" with MOMEL contour (d) and segmentation (c). Panes (a) and (b) represent the speech waveform and F_0 curve.

Although the beginning and end of the F_0 curve of Figure 7.2 are badly modelled by MOMEL, the middle part of the utterance is reasonably close to the F_0 curve. With a short utterance, of course, divergence at the two extremities is particularly noticeable. Figure 7.3 shows that this shortcoming applies to long utterances too; in most cases, though, the statistical impact would be negligible.

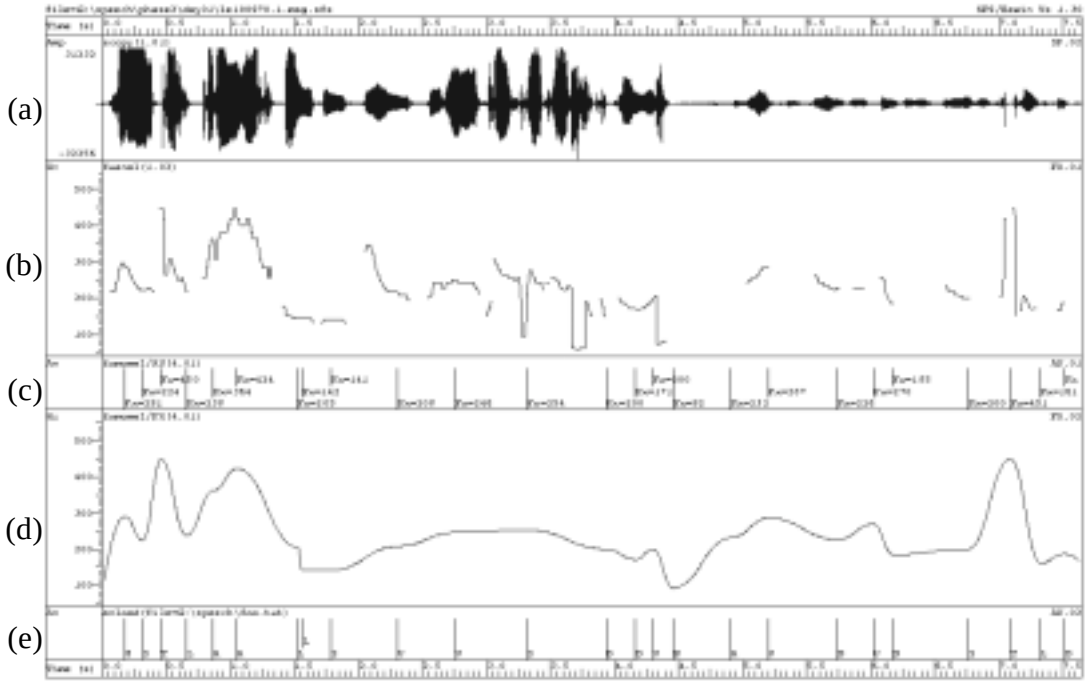

Figure 7.3 Longer utterance, with waveform (a), F_0 curve (b), MOMEL segmentation (c), MOMEL contour (d) and INTSINT labels (e). The labels (TPs) are given in small capitals in pane (e).

Looking at Figure 7.3, we can again see that the F_0 curve is well modelled by the MOMEL contour, and the placement of TPs at peaks and troughs in the contour can be observed. Some apparent halving errors have been suppressed, too, although regrettably this is not true of some instances of octave doubling: both TPs labelled T (top) by INTSINT seem erroneous.

Reading across the INTSINT labelling, notice that the first TP is labelled M. This is because only M, T and B are candidates for the initial TP, and the beginning of the utterance, naturally, is unlikely to carry extremes of the speaker's dynamic range. It is surprising, perhaps, that M, as the mean frequency, does not occur elsewhere in the utterance (this is true of utterances generally). There are two instances of S, although only one of these is correctly at approximately the same pitch as the preceding TP. The examples of U and D in this utterance represent, as expected, significantly smaller rises and falls from the preceding TP than do H and L respectively.

7.4 Experiments

The experiments were conducted along the same lines as for PLoNQ. First, different n-gram orders were explored.

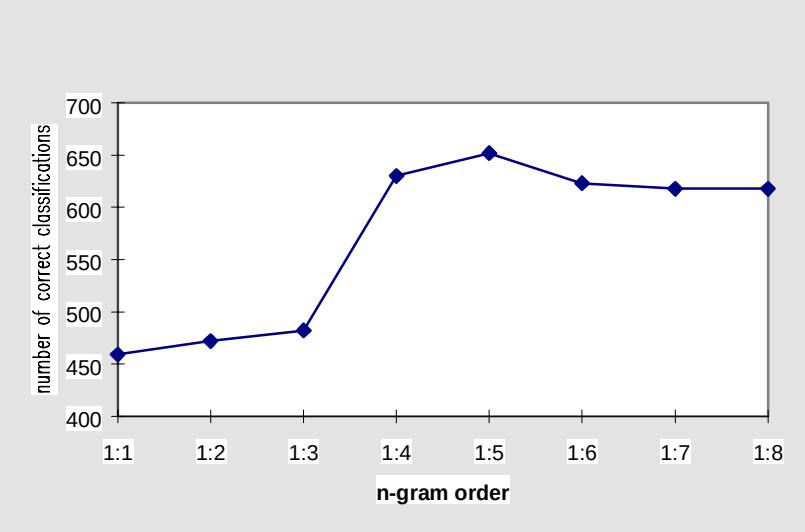

Figure 7.4 Classification performance using default parameters

Figure 7.4 shows the classification accuracy using INTSINT labels, for a range of n-gram orders. At order 1:5, there are 652 (32.78%) correct classifications. This is a significant finding, for it represents an improvement over the best score obtained with PLoNQ after parameter tuning, that is to say 645 (32.43%). Performance on the unigram model (order 1:1) also constituted an improvement on PLoNQ, where the best performance was less than 380 (19.11%) for all *clusnum* settings.

Experiments were conducted to establish the best value for *absfreq* (the minimum number of times a particular n-gram must occur in training data for it to be taken into account in the probability computation). Results are shown for higher n-gram orders in Figure 7.5.

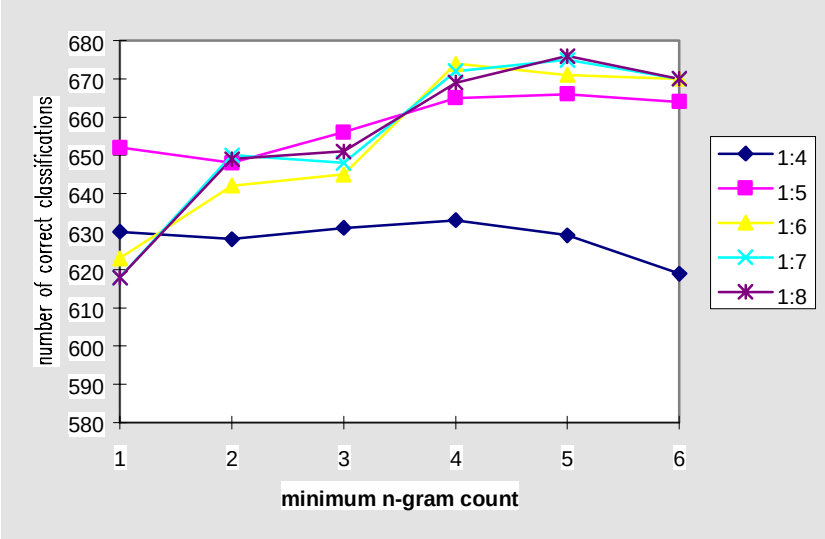

Figure 7.5 INTSINT performance at various values of *absfreq*

Figure 7.5 shows that the best performance is with *absfreq* set to 5 on a very high n-gram order. Under PLoNQ, the optimal value for *absfreq* was found to be 2; this discrepancy is readily explained by the fact that INTSINT segments are shorter, available labels are fewer, and therefore more n-gram tokens may generally be expected than with PLoNQ.

Experiments conducted up to this point have not exploited class prior information. Variation of the parameter *wprior*, which explicitly takes account of the training data incidence of each UT, yielded improved performance. Figure 7.6 shows the results for n-gram order 1:5 only, but the finding seen there, that *wprior* is optimally set to 0.1, in fact applies to all n-gram orders. The reader will recall from Chapter 5 that a value of 0 means the class prior is ignored, while 1 takes account of the prior probability without weighting.

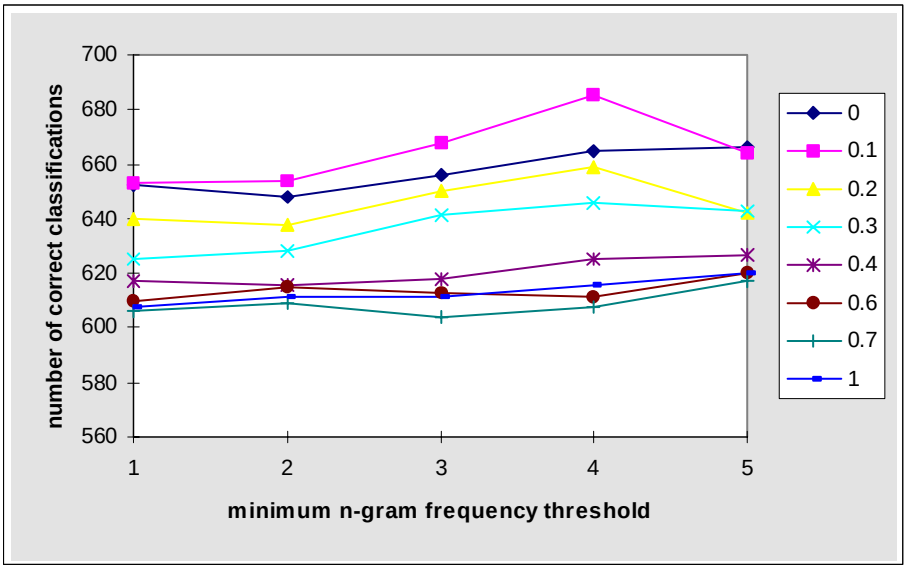

Figure 7.6 Performance at 1:5, *wprior* values ranging from 0 to 1

Figure 7.7 shows the performance at *wprior* = 0.1 for a range of n-gram orders and settings of *absfreq*.

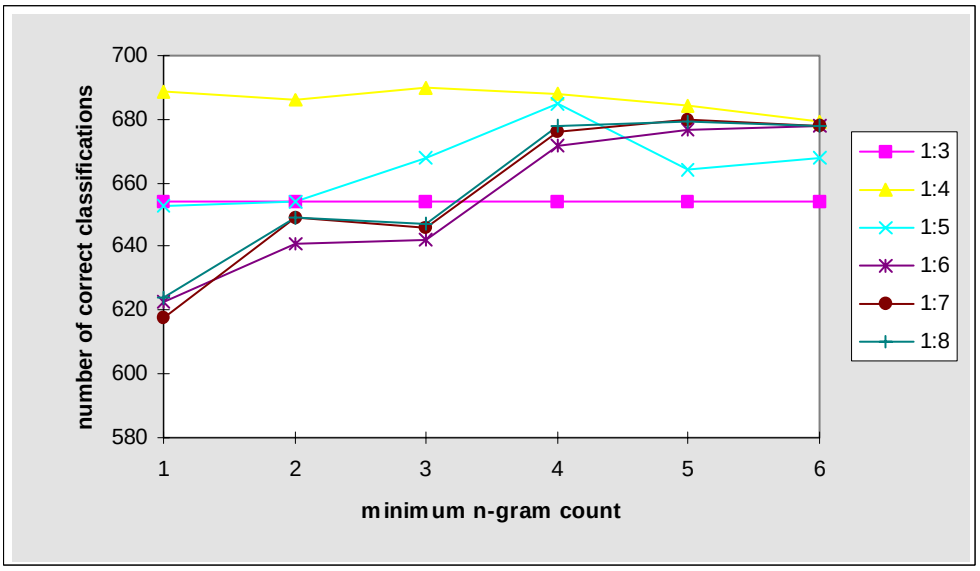

Figure 7.7 Performance with *wprior* set to 0.1

7.5 Concluding remarks

Although one good result was obtained with n-gram order 1:5, 685 (34.44%) correct classifications, it is clear from Figure 7.7 that the best performance was consistently on order 1:4. The top score attained, with of 3, was 690 (34.69%) correct classifications.

It is concluded, therefore, that the MOMEL and INTSINT stylization and discretization scheme is more successful at modelling prosody, for the purposes of UT classification, the linear trajectory and clustering approach of PLoNQ. Optimum classification performance under PLoNQ was 645 (32.43%), compared to 690 (34.69%) by the Hirst approach.

It was stated above that the Hirst approach has a greater degree of linguistic motivation than the entirely data-driven PLoNQ. It attempts, for example, to capture the overall prosodic shape of the utterance by seeking out the highest and lowest target points, as well as exploiting mean F_0 and dynamic range information. In the case of the relative symbols (H, L, U, D and S), the value of the previous target is the basis for TP computation: thus, attention is paid to the sequence of prosodic events at the labelling stage. In PLoNQ, the only treatment of prosodic event sequence (the n-gram model) takes place after the segment labelling has already been decided upon.

Furthermore, evidence from the class prior experiments suggests that the MOMEL and INTSINT features may be more powerful than those of PLoNQ. With the former, the best strategy is to rely on the classification given by the feature data, and not allow it to be overruled by the class prior, which is the effect of setting the parameter *wprior* to the low value of 0.1 that turned out to be optimal. In PLoNQ, on the other hand, more reliance was placed on the prior by setting *wprior* to a higher value, 0.875, in order to obtain optimal results.

8. Evaluation and conclusions

In this final chapter, the results of evaluative experiments using held-out data are presented. The significance of PLoNQ as a classification system is discussed, and limitations of the work are noted. Future possible research directions are also set out.

8.1 Held-out data experiments

The purpose of the experiments was explained in Chapter 1: because PLoNQ is a highly parametrized system, it was tuned to a set of operating points that either yielded the best performance with a specific test data set, or responded best to certain linguistic intuitions, particularly with respect to segment lengths. As part of the work reported in Chapter 5, multiple experiments were conducted to secure this optimal performance. However, if the system were ever to form part of a commercial application, it would clearly not be exposed over and over again to the same set of utterances; each 'live' utterance would be genuinely unseen and would not match exactly the prosodic profile of any in the test corpus. The use of held-out data (data which has not played any part in the experimentation so far) mirrors the use of real data in a production setting.

The first experiment was along the lines of the work of Chapter 5: the same primary and secondary training data sets were used, and the clustering and prosodic labelling did not need to be repeated. The 439 utterances of Oasis segment 9 were segmented and labelled according to the PLoNQ approach, sequences of labels were established as previously, and the new, unseen, utterances input to the classifier module. The optimal parameter values used were specified in Chapter 5, and are repeated in Table 8.1 for the reader's convenience.

Table 8.1 PLoNQ optimal parameter settings

module	parameter	available settings	optimal setting
pitch determination	algorithm used	cepstral auto-correlation integrated	integrated
pitch determination	pitch correction	binary	on
segmenter	duration penalty	variable	90000
segmenter	maximum segment length	variable	∞
segmenter	minimum segment length	variable	20 samples
clustering	number of centroids	variable	30
clustering	normalization	binary	off
clustering	secondary training	binary	on
n-gram model	n-gram order	variable	1:3
classifier	minimum n-gram occurrence	variable	2
classifier	class prior weighting	variable	0.875
classifier	floor value	variable	10^{-12}
classifier	mutual information	variable	$-\infty$
classifier	salience	variable	0

In this experiment, 134 utterances out of 439 were classified correctly, corresponding to 30.52%. This result is effectively equal to chance (30.75%) (that is, the number of utterances that properly belong to the largest class, *info*). The chance performance, obviously, would make use of class prior information; in this experiment the optimal class prior weight of Chapter 5, 0.875, was again used. It would almost certainly have been possible to achieve a better result by further parameter tuning, but this would have been against the spirit of the held-out experiments. The fact that this

result and that of the original experiment are at variance is best ascribed to a data sparseness problem: a difficulty with the held-out set is that it is rather small. It contains less than half the number of utterances of the other Oasis data segments, and the principal test set consisted of two such segments rather than one.

The second experiment used MOMEL and INTSINT, described in Chapter 7, instead of the first few modules of PLoNQ. As with the experiments reported in that chapter, prosodic labels were assigned by the algorithms designed by Hirst et al (2000), and this output was passed to the n-gram language model and classifier modules. By this means, 150 utterances were correctly classified, equivalent to an accuracy of 34.17%. Thus, the performance improvement by the Hirst approach over PLoNQ in these concluding experiments matched that reported for the principal data set experiments quite closely.

The results of the evaluative experiments, therefore, do reflect the general findings of the principal experiments. The result of the PLoNQ experiment is fairly close to chance, and the MOMEL and INTSINT segmentation and labelling again constituted a slight improvement on that performance.

8.2 Limitations of the work

Various limitations have been referred to in the course of the thesis. Perhaps the most crucial of these was the deterministic approach, or critical path navigation, necessitated by the parametrizable nature of the modules; it was pointed out in Chapter 5 that an exhaustive exploration of all permutations would have required a very large number of experiments. If PLoNQ were intended for immediate commercial exploitation, and substantial computer resources had been available, it would have been worthwhile (and indeed necessary) to explore all possibilities.

To illustrate: it was established in Chapter 5 that the optimal value for *wprior*, the class prior probability weighting parameter, was 0.875 for PLoNQ labelling, while given the MOMEL/INTSINT labelling described in Chapter 7, better performance was attained at *wprior* = 0.1. This discrepancy is attributable to the shorter utterance segments found by MOMEL. We might expect a different, perhaps improved, performance from PLoNQ if we were then to conduct further *ab initio* experiments, constraining segment durations further, and applying new settings of *wprior*. As noted, constraints of resources (and of time) meant that such experiments were not done.

Oasis is a well designed corpus, with a logical and coherent structure. In Chapter 3, though, it was pointed out that a small number of annotation errors had crept in. It is thought that the impact of these errors on the PLoNQ classification would have been minimal or non-existent, because nearly all of the relatively rare misclassifications involved class pairs that could not be differentiated on a prosodic analysis (such as *who* and *info*).

In the same chapter, it was noted that the annotation does not discriminate between the voices of men, women and children, although an annotation field is available for that purpose. The pitch detection algorithm used penalizes candidate samples outside the range 50 Hz to 500 Hz, but a switch is available which changes this range to 100-700 Hz, deemed more appropriate for higher voices. If the speaker status information had been available, and the higher range exploited, improved performance on both MOMEL and PLoNQ would possibly have been attained.

Recall from Chapter 3 that what has been referred to throughout the thesis as UT, utterance type, is described by BTexaCT (2001) as the **primary move type**. Each utterance is assigned not only one of these types, but also a task-specific **call class** such as *international directory enquiries* or *alarm call*. From an applications standpoint, it would have been preferable to make use of the latter categories in the experiments. Such an approach would not have been practical, though, because it simply would not lend itself to a prosodic analysis: it would be implausible to suggest that a particular pitch pattern could be associated with a request for an alarm call, for example. Furthermore, although there is an intuitive relationship between primary move types and the

illocutionary force types of speech act theory, described in Chapter 1, the prosody-based classification attempted in this thesis almost certainly makes greater appeal to attitudinal features of the utterance than the variation found in different IF types.

8.3 Directions for future research

The work of Gorin (Gorin 1995; Gorin et al 1999) does take a task-specific approach. This is possible because his analyses are lexical and (in experiments cited in Chapter 2) phone-based. Like the topic identification experiments reported in Chapter 2, they deal in keywords, or subparts of keywords.

The Oasis corpus includes a transcription of each utterance. If this were aligned with the sound file, there is no doubt that better performance could be obtained by combining the two data sources: in fact, the major contribution to classification would derive from the lexical model. Obviously, this experimental approach would not reflect a production classification problem, where no transcription is available unless it can be made in real time. Wright (1998) and King (1998) used prosodic features to reduce word error rate in a speech recognition system; potentially, a speech recognition module could be integrated with PLoNQ to improve its classification performance.

It was shown in Chapter 6 that different utterance types are, typically, of different lengths. Modelling utterance types on overall duration might therefore be a simple yet promising way to improve performance. Now, utterances of very different lengths tend to have different syntactic structures: short utterances may lack a classic subject-verb-object structure, longer utterances are likely to entail more complex structures such as relative clauses and long-distance dependencies. Given this, it is plausible that different UTs may exhibit particular syntactic dependencies; certainly, for example, one might reasonably expect a different structure from a request for information, which might come in the form of a question, and a problem statement. Given suitable tagging and parsing tools, it might well be possible to recruit a syntactic contribution to UT classification.

Finally, it would probably be of benefit to repeat some of the PLoNQ and MOMEL/INTSINT experiments using a different configuration of UTs. In Chapter 3, for example, it was shown that the class *who* is effectively a subset of class *info*, and that the distinctions between the two could not be prosodically modelled; the *connect* and *action* classes are in the same relationship. If these pairs of classes were conflated, performance benefits would be likely to ensue. Experiments with other class combinations could also be attempted.

8.4 Summary

This thesis has presented data-driven techniques for segmenting utterances drawn from a corpus of spontaneous speech. Each segment was assigned a prosodic label, according to its gradient, average fundamental frequency and duration, by K-means clustering. The utterances were also labelled, in a separate experiment, with the INTSINT prosodic coding. Contiguous labels were then assembled into sequences, following Gorin et al (1999), and the resulting n-grams used to probabilistically assign similarly labelled unseen utterances to one of six classes determined by the annotators of the Oasis corpus.

In a psycholinguistic experiment, reported in Chapter 6, it was shown that human subjects are able to classify utterances by prosodic content only; in fact, the participants in the experiment outperformed the automatic system by a considerable margin, choosing the correct category 53.16% of the time. This result was arrived at using a weighted scoring system that took into account participants' skill at classifying the utterance when the non-prosodic content of the utterance was not suppressed.

By means of comparative experiments with the two automatic labelling systems, and a good deal of parameter adjustment, it was established that 34.69% of utterances in an evaluation set, consisting of truly unseen data, could be correctly classified; the probability of correct classification by chance was 30.75%. This result was obtained by combining MOMEL stylization and INTSINT discretization with the PLoNQ label sequencing and classifier modules. A similar improvement on chance was obtained with the principal test set, demonstrating that the performance would generalize to genuine unseen data.

The PLoNQ system of segmenting and labelling utterances was not particularly successful. This is probably because of the lack of linguistic motivation behind it. INTSINT labelling takes a holistic approach to the utterance, locating the highest and lowest points where voicing is present, and using this as the basis of the segmentation. PLoNQ labelling, on the other hand, is conducted in a standalone fashion, simply locating the centroid that is closest to the segment under analysis.

Although performance is far from perfect, the findings are of particular interest because the system is for the most part data-driven. This is not to say that the system is devoid of heuristic procedures: it has been made clear in the course of the thesis that linguistically motivated parameter tuning was a central theme of the work. Nevertheless, the system is entirely automated, and no prosodic transcription or hand annotation of any kind is necessary.

The implementation of this project was dependent on Hirst's (2000) MOMEL and INTSINT algorithms. As far as can be ascertained, this thesis constitutes the first piece of research to exploit these algorithms in a classification task. Although Hirst's work includes stringent evaluation procedures, these have previously been confined to re-synthesis of the discretized contour, and comparison with the raw F_0 curve. In this respect, PLoNQ makes a significant contribution to research in the domains of prosody and utterance classification.

The UT classification techniques proposed could be readily extended to other multi-class problems, such as speaker recognition, spoken language identification and topic identification in audio data. Integrated with a data contribution from other linguistic strata, such as lexis and syntax, PLoNQ would provide the research community with an invaluable tool for preliminary class annotation, and could potentially be extended to a variety of commercial applications.

Bibliography

Abercrombie, D (1967) *Elements of general phonetics*. Edinburgh University Press

Academia Sinica (1998) Sinica Balanced Corpus, available at http://www.sinica.edu.tw/ftms-bin/kiwi.sh

Anderson, A, M Bader, E Bard, E Boyle, G Doherty, S Garrod, S Isard, J Kowtko, J McAllister, J Miller, C Sotillo, H Thompson & R Weinert (1991) The HCRC Map Task Corpus. In *Language and Speech*, 34(4):351-366

Austin, J (1962) *How to do things with words*. Oxford: Clarendon

Bacchiani, M (2000) Using maximum likelihood linear regression for segment clustering and speaker identification. In *Proc ICSLP*, Beijing, 536-539

Bagshaw, P (1994) Automatic prosodic analysis for computer-aided pronunciation teaching. PhD thesis, University of Edinburgh

Beauchaine, T & R Beauchaine (2002) A comparison of maximum covariance and k-means cluster analysis in classifying cases into known taxon groups. In *Psychological Methods*, 7: 245-261

Bengio, Y (1996) *Neural networks for speech and sequence recognition*. London: International Thomson Computer Press

Bird, S, S Browning, R Moore & M Russell (1995) Dialogue move recognition using topic spotting techniques. In *Proc ESCA Workshop on Spoken Dialogue Processing - Theory and Practice*, Vigsø, Denmark

Breiman, L, R Friedman, R Olshen & C Stone (1984) *Classification and regression trees*. Pacific Grove, CA: Wadsworth

BTexaCT (2001) OASIS First Utterance corpus, version 2.23 (annotated transcription, audio files and release notes)

Bwantsa-Kafungu, S (1972) *J'apprends le lingala tout seul en trois mois*. Kinshasa: Centre de recherches pedagogiques

Campione, E & J Véronis (2001) Semi-automatic tagging of intonation in French spoken corpora. In *Proc Corpus Linguistics*, Lancaster, 90-99

Campione, E (2001) Etiquetage prosodique semi-automatique de corpus oraux: algorithmes et méthodologie. PhD thesis, Université de Provence

Carey, M, E Parris, H Lloyd-Thomas & S Bennett (1996) Robust prosodic features for speaker identification. In *Proc ICSLP*, Philadelphia, 1800-1803

Couper-Kuhlen, E (1986) *An introduction to English prosody*. London: Arnold

Crystal, D (1969) *Prosodic systems and intonation in English*. Cambridge University Press

Dunning, T (1994) Statistical identification of language. Technical report CRL MCCS-94-273, Computing Research Lab, New Mexico State University. Available at http://www.comp.lancs.ac.uk/computing/users/paul/ucrel/papers/lingdet.ps

Farinas, J (1999) La prosodie pour l'identification automatique des langues. PhD thesis, Institut de Recherche en Informatique de Toulouse

Fox, A (1984) Subordinating and co-ordinating intonation structures in the articulation of discourse. In D Gibbon & H Richter (eds) 120-133

Fromkin, V (1978) (ed) *Tone: a linguistic survey*. New York: Academic Press

Garner, P & A Hemsworth (1997) A keyword selection strategy for dialogue move recognition and multi-class topic identification. In *Proc ICASSP*, Munich, 1823-1826

Garner, P (1997) On topic identification and dialogue move recognition. In *Computer Speech and Language* 11(4): 275-306

Gibbon, D & H Richter (eds) *Intonation, accent and rhythm: studies in discourse phonology*. Berlin: de Gruyter

Godfrey, J, E Holliman & J McDaniel (1992) SWITCHBOARD: Telephone speech corpus for research and development. In *Proc ICASSP*, San Francisco, 517-520

Gong, X & M Richman (1995) On the application of cluster analysis to growing season precipitation in North America east of the Rockies. In *Journal of Climate*, 8: 897-931

Gorin A, D Petrovska-Delacrétaz, G Riccardi & J Wright (1999) Learning spoken language without transcriptions. In *Proc Workshop on Automatic Speech Recognition and Understanding*, Keystone CO

Gorin, A (1995) On automated language acquisition. In *Journal of the Acoustical Society of America*, 97-6: 3441-3461

Grabe, E, F Nolan & K Farrar (1998) IViE - A Comparative Transcription system for Intonational Variation in English. In *Proc ICSLP*, Sydney, 1259-1262

Hartigan, J (1975) *Clustering algorithms*. New York: Wiley

Hemphill, C, J Godfrey & G Doddington (1990) The ATIS Spoken Language Systems pilot corpus. In *Proc DARPA Speech and Natural Language Workshop*, Hidden Valley PA, 96-101

Hess, W (1983) *Pitch determination of speech signals*. New York: Springer-Verlag

Hirschberg J, D Litman, M Swerts (1999) Prosodic cues to recognition errors. In *Proc Workshop on Automatic Speech Recognition and Understanding*, Keystone, Colorado.

Hirst, D, A Di Cristo & R Espesser (2000) Levels of representation and levels of analysis for the description of intonation systems" in M Horne (ed) *Intonation: theory and experiment*, 51-87. Dordrecht: Kluwer

Huckvale, M & University College London (2001) Speech Filing System. Available at http://www.phon.ucl.ac.uk/resource/sfs

Jensen, U, R Moore, P Dalsgaard & B Lindberg (1994) Modelling intonation contours at the phrase level using continuous density hidden Markov models. In *Computer Speech and Language* 8: 247-260

Johns-Lewis, C (1986a) (ed) *Intonation in discourse*. London: Croom Helm

Johns-Lewis, C (1986b) Prosodic differentiation of discourse modes. In Johns-Lewis (ed), 199-220

Johnson, S (1993) Solving the problem of language recognition. Technical report, School of Computer Studies, University of Leeds

Jurafsky D, R Bates, N Coccaro, R Martin, M Meteer, K Ries, E Shriberg, A Stolcke, P Taylor & C Van Ess-Dykema (1997) Automatic detection of discourse structure for speech recognition and understanding. In *Proc Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, 88-95

Kerbrat-Orecchioni, C (1977) *La connotation*. Presses Universitaires de Lyon

King, S (1998) Using Information Above the Word Level for Automatic Speech Recognition. PhD thesis, University of Edinburgh

Kobayasi,Y, T Takunaga & H Tanaka (1994) Analysis of Japanese Compound Nouns using Collocational Information. In *Proc Coling*, Tokyo, 865-869

Krause, M (1984) Recent developments in speech signal pitch extraction. In D Gibbon & H Richter (eds), 243-252

Ladd, D, K Scherer & K Silverman (1986) An integrated approach to studying intonation and attitude. In Johns-Lewis (ed), 125-138

Ladefoged, P (1962) *Elements of Acoustic Phonetics*. University of Chicago

Lager, T, N Zinovjeva (1999) Training a dialogue act tagger with the µ-TBL system. In *Proc Third Swedish Symposium on Multimodal Communication*, Linköping, Sweden

Lauer, M (1995) Corpus statistics meet the noun compound: some empirical results. In *Proc ACL*, Cambridge MA, 47-54

Laver, J (1994) *Principles of phonetics*. Cambridge University Press

Lazzari, G, R Frederking, W Minker (1999) Speaker-language identification and speech translation. In *Multilingual information management: current levels and future abilities.* Pittsburgh: Carnegie Mellon University Computer Science Dept.

Levin, H, C Schaffer, & C Snow (1982) The prosodic and paralinguistic features of reading and telling stories. In *Language and Speech*, 25: 43-54

Lieberman, P (1967) *Intonation, perception and language*. Cambridge, MA: MIT Press

Linguistic Data Consortium (1993) ATIS complete corpus, available at http://www.ldc.upenn.edu

Markel, J & A Gray (1976) *Linear prediction of speech*. New York: Springer

Medan, Y, E Yair & D Chazan (1991) Super resolution pitch determination of speech signals. In *IEEE Trans. ASSP*, 39(1): 40-48

Nagata, M & T Morimoto (1993) An experimental statistical dialogue model to predict the speech act type of the next utterance. In K Shirai, T Kobayashi & Y Harada (eds) *Proceedings of the International Symposium on Spoken Dialogue*, 83-86.

Nakai, M, H Singer, Y Sagisaka & H Shimodaira (1996) Accent phrase segmentation by F0 clustering using superpositional modelling. In Sagisaka et al (eds)

Noguchi, H, K Kiriyama, H Matsuda, M Taniguchi, Y Den & Y Katagiri (1999) Automatic labeling of Japanese prosody using J-ToBI style description. In *Proc Eurospeech*, Budapest, 2259-2262

Noll, A (1967) Cepstrum pitch determination. In *Journal of the Acoustical Society of America*, 41, 293-309

Nöth, E, A Batliner, V Warnke, J Haas, M Boros, J Buckow, R Huber, F Gallwitz, M Nutt & H Niemann (1999) On the use of prosody in automatic dialogue understanding, in *Proc ESCA Workshop on Dialogue and Prosody*, 25-34, Eindhoven

Oakes, M (1998) *Statistics for Corpus Linguistics*. Edinburgh University Press

O'Connor, J & G Arnold (1961) *Intonation of colloquial English*. Harlow: Longman

Ohala, J (1978) *The production of tone*. In V Fromkin (ed), 5-39

Pickett, J (1980) *The sounds of speech communication*. Baltimore: University Park Press

Pierrehumbert, J (1980) The phonology and phonetics of English intonation. PhD thesis, MIT

Pijper, J (1983) *Modelling British English intonation*. Dordrecht: Foris

Pike, K (1948) *Tone languages*. Ann Arbor: University of Michigan Press

Rapp, S (1998) Automatic Labelling of German Prosody. In *Proc ICSLP*, Sydney, 1267-1270

Roach, P (1991) *English phonetics and phonology: a practical course*. Cambridge University Press

Ross, M, H Shaffer, A Cohen, R Freudberg & H Manley (1974) Average magnitude difference function pitch extractor. In *IEEE Trans. ASSP*, 22, 353-362

Sagisaka, Y, N Campbell & N Higuchi (eds) (1996) *Computing prosody: computational models for processing spontaneous speech*. New York: Springer

Samuel, K, S Carberry & K Vijay-Shanker (1998) Dialogue act tagging with transformation-based learning. In *Proc 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, 1150-1156.

Scheffers, M (1988) Automatic stylization of F0 contours. In *Proc Speech 88*, Edinburgh, 981-988

Searle, J & D Vanderveken (1985) *Foundations of illocutionary logic*. Cambridge University Press

Searle, J (1976) A taxonomy of illocutionary acts. In *Language in Society*, 5: 1-23

Secrest, B & Doddington, G (1983) Integrated Pitch Tracking Algorithm for Speech Systems. In *Proc ICASSP*, Boston, 1352-1355

Shriberg, E, R Bates, P Taylor, A Stolcke, K Ries, D Jurafsky, N Coccaro, R Martin, M Meteer & C Van Ess-Dykema (1998) Can prosody aid the automatic classification of dialog acts in conversational speech? In *Language and Speech* 41: 443-492

Smith, S & M Russell (2001) Determining query types for information access. In *Proc Corpus Linguistics*, Lancaster, 562-570.

Smith, S (1999) Discontinuous compounds in Mandarin Chinese: a lexicalization algorithm. MSc thesis, UMIST, Manchester

Taylor, P (2000) Analysis and synthesis of intonation using the Tilt model. In *Journal of the Acoustical Society of America*, 107-3:1697-1714

'tHart, J (1991) F0 stylization in speech: straight lines versus parabolas. In *Journal of the Acoustical Society of America* 6: 3368-3370

Thymé-Gobbel, A & S Hutchins (1996) On using prosodic cues in automatic language identification. In *Proc ICSLP*, Philadelphia, 1768-1771

Wang, C & S Seneff (2001) Prosodic scoring of recognition outputs in the JUPITER domain. In *Proc Workshop on Prosody in Speech Recognition and Understanding*, Red Bank, NJ, 151-156

Wightman, C (2002) ToBI Or Not ToBI? In *Proc Speech Prosody*, Aix-en-Provence

Williams, B (1986) An acoustic study of some features of Welsh prosody. In Johns-Lewis (ed), 35-52

Wright, H (1998) Automatic Utterance Type Detection Using Suprasegmental Features. *Proc ICSLP*, Sydney, 1403-1406

Wright, H (2000) Modelling Prosodic and Dialogue Information for Automatic Speech Recognition. PhD thesis, University of Edinburgh

Wright, J, M Carey & E Parris (1995) Improved topic spotting through statistical modelling of keyword dependencies. In *Proc ICASSP*, Detroit, 313-316

Ying, G, L Jamieson & C Mitchell (1996) A probabilistic approach to AMDF pitch detection. In *Proc ICSLP*, Philadelphia, 1201-1204

Zhang, T, R Ramakrishnan & M Livny (1997) Birch: A new data clustering algorithm and its applications. In *Data Mining and Knowledge Discovery*, 1(2): 141-182