Prosody Resources and Symbolic Prosodic Features for Automated Phrase Break Prediction

Claire Brierley

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy

The University of Leeds
School of Computing

September 2011

The candidate confirms that the work submitted is her own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

It is universally recognised that humans process speech and language in chunks, each meaningful in itself. Any two renditions or assimilations of a given sentence will exhibit similarities and discrepancies in chunking, where speakers and readers use pauses and inflections to mark phrase breaks. This thesis reviews deterministic and stochastic approaches to phrase break prediction, plus datasets, evaluation metrics and feature sets. Early rule-based experimental work with a chunk parser gives rise to motivational insights, namely: the limitations of traditional features (syntax and punctuation) and deficiency of prosody in current phrasing models, and the problem of evaluating performance when the training set only represents one phrasing variant. Such insights inform resource creation in the form of ProPOSEL, a ***pro***sody and ***p***art-***o***f-***s***peech ***E***nglish ***l***exicon, to create a domain-independent knowledge source, plus prosodic annotation and text analytics tool for corpus-based research, supported by a comprehensive software tutorial. Future applications of ProPOSEL include prosody-motivated speech-to-viseme generation for 'talking heads' and expressive avatar creation. Here, ProPOSEL is used to build the ProPOSEC dataset by merging and annotating two versions of the Spoken English Corpus. Linguistic data arrays in this dataset are first mined for prosodic boundary correlates and later re-conceptualised as training instances for supervised machine learning. This thesis contends that native English speakers use certain sound patterns (*e.g.* diphthongs and triphthongs) as *linguistic signs* for phrase breaks, having observed these same patterns at rhythmic junctures in poetry. Pre-boundary lexical items bearing these complex vowels and gold-standard boundary annotations are found to be highly correlated via the chi-squared statistic in different genres, including seventeenth century English verse, and for multiple speakers. Complex vowels and other symbolic prosodic features are then implemented in a phrasing model to evaluate efficacy for phrase break prediction. The ultimate challenge is to better understand how sound and rhythm, as components of the linguistic sign, inform psycholinguistic chunking even during silent reading.

Declaration

I declare that the work presented in this thesis is original, to the best of my knowledge in this research field, and that it is my own work. Much of this thesis has already been published in the following papers. Section references at Level 2 (*e.g.* 8.9) mean that the entire section is taken from the paper; further specification at Level 3 (*e.g.* 5.7.1) indicates that only that sub-section is taken from the paper.

Brierley, Claire; Atwell, Eric. Prosodic phrase break prediction: problems in the evaluation of models against a gold standard. Traitement Automatique des Langues, vol. 48.1. 2008. (Chapters 1, 3, 4 and 5: 1.2; 3.1 to 3.5; 4.3.3; 5.3)

Brierley, Claire; Atwell, Eric. An approach for detecting prosodic phrase boundaries in spoken English. ACM Crossroads, vol. 14.1. 2007. (Chapters 1 and 4: 1.6; 4.2)

Brierley, C. and Atwell, E. 2009. Exploring Phrase Break Correlates in a Corpus of English Speech with ProPOSEL, a Prosody and POS English Lexicon. In proceedings of Interspeech 2009 in Brighton. September 2009. (Chapters 3, 5 and 8: 3.7; 5.6 and 8.3 to 8.5)

Brierley, C. and Atwell, E. 2010. Complex Vowels as Boundary Correlates in a Multi-Speaker Corpus of Spontaneous English Speech. In proceedings of Speech Prosody 2010, Chicago. May 2010. (Chapters 5 and 8: 5.2; 8.7 and 8.8)

Brierley, Claire; Atwell, Eric. ProPOSEL: a human-oriented prosody and POS English lexicon for machine learning and NLP in: Proceedings of International Conference on Computational Linguistics (CoLing 2008), Workshop on Cognitive Aspects of the Lexicon. Manchester. August 2008. (Chapter 6: 6.7.2 to 6.8)

Brierley, C. and Atwell, E. 2010. Holy Smoke: Vocalic Precursors of Phrase Breaks in Milton's 'Paradise Lost'. In Journal of Literary & Linguistic Computing, Volume 25, Issue 2. (Chapter 7: 7.4 to 7.10)

Brierley, C. and Atwell, E. 2010. ProPOSEC: a Prosody and PoS Spoken English Corpus. In proceedings of LREC 2010, Valetta, Malta. May 2010. (Chapter 8: 8.9 and 8.10)

Brierley, C and Atwell, E. 2011. Non-traditional prosodic features for automated phrase break prediction. In *Literary and Linguistic Computing, doi: 10.1093/llc/fqr023.* (Chapter 10: 10.9.1)

A further paper has been submitted (but not published) and forms the basis of Chapter 5 of the thesis.

Brierley, C. and Atwell, E. 2010. Building ProPOSEL: a Prosody and POS English Lexicon for Machine Learning, Text Analytics and Stylometry. Submitted to Language Resources and Evaluation Journal. (Chapter 6: 6.2 to 6.7.1)

Dedication and Acknowledgements

This thesis is dedicated to my two sons, Max and Louis. One reason I started it was to give myself something meaningful to do once my sons had left home – but that plan didn't really work because I'd got about half way through and they were still there! Thanks to both of you, Max and Louis, for talking me through those times when I got despondent and self-doubting. And thanks for your company always.

Sincere thanks also to Dr Andrew Hartley, my then Head of School, and long-term colleague at the University of Bolton, for supporting this endeavour at the outset, along with Bruce Cain, Director of Human Resources. I'm grateful to both of you for giving me this chance, and for your empathy as managers. I also wish to thank Dr Chris Minta for your comradeship and support at Bolton, along with Shirley Silcock.

I have a special mention for two colleagues in Computing at Bolton, Peter Lager and Dr Steve Manning. I felt able to approach both of you for advice: Peter on asynchronous lists in Deane Deli, and Steve on chi-squared. On the latter, it was such a relief to have someone agree with me that the text book I was consulting was pretty indecipherable and wasn't it better to look at worked examples in an early edition of *Statistics for the Terrified*!

Another person I felt able to approach for help right at the beginning was Paul Querelle. You helped me first install NLTK (an earlier, trickier version); but I guess the most important thing you did was simply to confirm that my initial (and, as I then thought, silly) idea for extracting text from Aix-MARSEC TextGrid files wasn't silly at all; it was in fact the right approach – *because that's what programmers do*!

Dr Eric Atwell as supervisor often gets the *last but not least* slot in PhD acknowledgements, but not in this case. I remember signing up for the PhD and getting an email back from Eric saying: "Welcome to the PhD challenge! ☺" That was something of a Laurel and Hardy moment, where I played both characters: "Claire," (scratching my head), "now look what you've gotten me into!" Eric also gave me a 'real' thesis from one of his ex-students to look at right back at the beginning (I’ve still got it, like a talisman), and during one early conversation (involving lots of walking up and down steps round the campus), Eric said that when things got tough, I should never forget my original motivation for embarking on the PhD. I think these anecdotes give some idea of some of Eric’s strengths as a supervisor. I am not the first person to say that Eric has a very good understanding of people and what motivates them. For me, his style of supervision has been the right one: it gives the student freedom, it is positive and encouraging, and I’ve come out of this with lots of publications, an achievement I really value.

Sincere thanks also to Dr Katja Markert in Computing at Leeds: whenever we’ve had a tutorial, I’ve always come away having substantially learnt something, and with a way forward, particularly on evaluation.

Thanks also to my colleagues and fellow students in the NLP group at Leeds, especially for sharing ideas and feedback on dry-run conference presentations. Thanks especially to Dr Clive Souter, Justin Washtell, and Owen Nancarrow for moral support extended when I most needed it.

Finally, the biggest ever thankyou to Majdi Sawalha in Arabic NLP here at Leeds (and back in Jordan), because again, without your steadfast moral support, I would have found it very difficult to complete the latter stages of this PhD. Majdi, I really value our friendship and look forward to “the great escape” when we have both finished our PhDs and can pursue all our new research ideas. I wish you all the very best in your own immediate PhD undertaking, and for your future career and happiness.

Contents

Figures

Text for some captions may be summarised here to save space.

Tables and Code Listings

Text for most table captions is summarised here to save space.

Code Listings

Text for most code listings is summarised here to save space.

Glossary of Main Terms used in Thesis

Aix-MARSEC	Most recent version of MARSEC corpus, with multiple prosodic annotation tiers
ANA	Anacrusis or unstressed syllable(s)
ARPAbet	Subset of IPA phonetic transcription scheme for American English
ASR	Automatic Speech Recognition
BCR	Balanced classification rate
Boundary	Prosodic annotation mark denoting intonational phrase structure in English. Terminology synonymous to this definition of boundary and also used in this thesis is: tone group boundary; tone unit boundary; phrase break.
BNC	British National Corpus
C5	CLAWS 5 BNC tagset
C7	CLAWS 7 tagset
CELEX-2	Lexical database for English, German and Dutch (version 0.2)
CFP algorithm	Content/function word + punctuation algorithm
Chink-chunk	Rule that assigns phrase breaks whenever a content word or *chunk* is immediately followed by a function word or *chink*
Chunk	Prosodic-syntactic unit manifest as (in most cases) a sequence of words which, even if taken out of context, would retain/convey integral meaning
CLAWS	Constituent Likelihood Automatic Word Tagging System
Complex vowels	The diphthongs and triphthongs of Received Pronunication (RP) or BBC English (*cf.* Table 7.5)
CMU	Carnegie-Mellon Pronouncing Dictionary
CUV2	Computer-usable dictionary (version 0.2)
CUVPlus	Enhanced computer-usable version of OALD
CV pattern	Consonant-vowel pattern
DISC	Distinct Single Characters phonetic transcription scheme
EModE	Early Modern English
FN	False negative
FP	False positive
IOB	Inside-outside-beginning tagging scheme for shallow parsing
IPA	International Phonetic Alphabet
IU	Intonation unit
LDOCE	Longman Dictionary of Contemporary English
LOB	Lancaster-Oslo-Bergen tagset
MARSEC	Machine Readable Spoken English Corpus
NLTK	Natural Language ToolKit
NRU	Narrow Rhythm Unit
OALD	Oxford Advanced Learner's Dictionary
Penn	Penn Treebank tagset
POS	Part-of-speech
PresE	Present-day English
ProPOSEC	Prosody and part-of-speech [annotated version of] SEC
ProPOSEL	Prosody and part-of-speech English lexicon
SAM-PA	Speech Assessment Methods Phonetic Alphabet
SEC	Spoken English Corpus
TN	True negative

ToBI	Tones and Break Indices prosodic annotation scheme
TP	True positive
TTS	Text-to-Speech [Synthesis]
UCREL	[Lancaster] University Centre for Computer Corpus Research on Language
VTTS	Visual Text-to-Speech [Synthesis]

Chapter 1
Introduction

1.1. Thesis overview

The application area pertinent to this thesis is automated phrase break prediction, an NLP task within the TTS pipeline which sub-divides input text into meaningful units, phrases or chunks below sentence level to re-enact as closely as possible the way in which an articulate native speaker (or reader) might chunk (or parse) the utterance to maximise communication effectiveness (or understanding). In an automated system, predicting phrase breaks is synonymous with classifying junctures between words, or the words themselves, as either breaks or non-breaks. Once these break points or boundary delimiters have been discovered, intervening text can then be further 'animated' with prosody, for example, it can be given a suitable intonation contour. Due to the modular TTS architecture, phrase break classifiers assume prior sentence segmentation and part-of-speech tagging for input text. Thus, punctuation and syntax are traditionally used as predictive features during classification. This thesis sets out to discover additional, *prosodic* phrase break correlates which may be used in conjunction with traditional features to enhance classifier performance. One of the artefacts produced in this thesis is ProPOSEL, a prosody and part-of-speech English lexicon for annotating the words of a text with an array of linguistic attributes (phonetic, prosodic and syntactic) which can be further transformed into candidate discriminatory classification features. The predictive potential of several transformations of this kind, including the principal thesis finding of complex vowels (*i.e.* the diphthongs and triphthongs of Received Pronunciation or BBC English) as proven phrase break correlates, is evaluated on a custom-built dataset via a simple phrase break model.

1.2. Prosodic phrasing

Prosodic phrasing is a universal characteristic of language (Ladd, 1996) and refers to the way speakers of any given language process speech as a series of chunks: meaningful, stand-alone clusters of words which have some relationship to syntactic phrase structure, the 'natural joints' in sentences (Abney, 1995). For example, Croft (1995) found that 97% of prosodic units in a corpus of English oral narratives were also syntactic units. The correlation and discrepancy between prosody and syntax is a continuing debate in the literature, but there does appear to be consensus on the fact that prosodic phrasing is simpler, shallower and flatter than syntactic structure. Abney (1992) proposes the unifying concept of *performance structure*, the way in which prosody and syntax interact in practice.

Performance structure in English is realised and perceived as a partnership between pitch accents and pauses which draws attention to these natural joints or boundaries in the speech stream.

> '...The correct question about sentence accent data is not 'Why is the main prominence in this sentence on word X rather than on word Y?' but rather 'Why is this sentence divided up into phrases the way it is?' (Ladd, 1996, p.196; *ibid. cf.* p.233).

The goal of automatic phrase break prediction is, therefore, to identify natural joints in *text* which correspond naturally and intelligibly (these are the important criteria) to the way a native speaker might process or chunk that same text as speech. In text, prominent boundaries are marked by punctuation and it is second nature for us to associate different intonation and different degrees of pause with the various punctuation marks when reading that text aloud. Thus language models designed to predict prosodic phrase breaks from input text – for Text-to-Speech Synthesis (TTS) applications, for example – will often use punctuation as a primary cue.

Prosodic phrasing and intonation exhibit a dual purpose in speech: a chunking function to identify meaningful – and syntactically coherent – clusters of words and a highlighting function to emphasise salient items within clusters. In English, chunking and highlighting are often conflated (Peppe, 2006): prominent words tend to complete a phrase group and so occupy pre-boundary position. The convergence and *non*-convergence of these functions has consequences for the evaluation of language models that try to simulate them.

1.3. Phrasing and punctuation

The previous section refers to the debate about correspondence and anomaly at the prosody-syntax interface. Empirical evidence for this dichotomy is examined in detail in Chapter 5 of this thesis. The status of punctuation as prosodic boundary marker is also problematic. On the one hand, punctuation is a very reliable boundary predictor, capturing about 50% of phrase breaks in text as evidenced in corpus-based studies (Taylor, 1996, p.131) and via experimentation (Taylor and Black, 1998; Ingulfsen, 2005). However, the performance of a classifier is in part measured by the number of correct phrase breaks it recaptures in a sample of unseen text stripped of its original boundary locations. Thus, while punctuation aids *precision* in that it does not lead to over-generation or insertion of boundaries where they are not supposed to be, it falls short on *recall* since we can expect twice as many boundaries as punctuation marks. A classifier therefore needs additional clues, available to humans performing such a task, to inform phrase break assignment:

> '…As a rule of thumb, when we read a passage aloud, we are likely to use spoken boundaries corresponding to punctuation marks, and we have to decide where other, additional boundaries should be placed…' (Quirk *et al.* in Taylor, 1996, p.130).

1.4. What is a chunk?

So far, we have loosely defined chunks as sequences of words that make sense as a stand-alone unit but which do not generally constitute a conventional written sentence (unless that sentence is very short). We have noted their resemblance to syntactic units, for example noun phrases or sentence clauses, and we have noted the correspondence between intelligent chunking and punctuation as chunk delimiter. Datasets used in this thesis merge information from different versions of the Spoken English Corpus (SEC) and carry their own attendant definition of what constitutes a chunk. This is first and foremost a *prosodic* unit. Such units are variously referred to as tone units or intonation groups or tone groups or intonational phrases in the literature (Croft, 1995). To qualify as a chunk by this definition, the posited sequence must include at least one accented word, namely, a word that exhibits pitch variation on the stressed syllable in the listener's/transcriber's perception (Dehe and Wichman, 2010). SEC identifies two levels of chunk via two different boundary markers: the tone unit boundary (|) and the pause (||).

At this point, we mention additional prosodic terminology to do with prominence and intonation. The former is a property of syllables which are perceived as being louder and longer than others and which may enact changes in pitch. The latter is generally a property of the phrase or sentence – although in emphatic or emotional speech, individual words may exhibit a complete intonation contour (Welby, 2003) – and refers to the tune of an utterance: recognisable patterns in a series of pitch movements which have semantic and functional significance. In English, prominent syllables can be stressed, in which case they are perceived as strong rhythmic beats endowed with a full vowel, not a reduced one. Stressed syllables may also be accented, in which case they initiate a change in the direction of pitch and sometimes a sharp jump across the speaker's pitch range. British English has six distinct pitch accent types: rising; falling; rising-falling; falling-rising; rising-falling-rising; and level (Grabe, 2001), where the first four are considered to be major types (Dehe and Wichman, 2010).

1.4.1. A Semiotics perspective on chunks

Chunks are psychological as well as physical entities. One property of the linguistic sign as defined by Saussure is linearity (Saussure in Holdcroft, 1991, p. 52), such that syntagmatic relations exist between one chunk and another, since the notion of *syntagm* is not simply restricted to word level units in a language:

> '…the linguistic entity is not accurately defined until it is delimited, i.e. separated from everything that surrounds it on the phonic chain. These delimited entities or units stand in opposition to each other in the mechanism of language…' (*ibid*. p.89)

Furthermore, the multiplicity of permissible chunk combinations and sequences is an aspect of the productive or generative nature of language. The particular focus of this thesis is on *prosodic* delimiters between contiguous chunks.

1.4.2. Language generation and phrase construction within a cognitive framework

Psycholinguists have investigated language generation and phrase construction within a cognitive framework, and their conclusions are relevant for this research. Kempen and Hoenkamp (1982) observe that:

> '...Human speakers often produce sentences incrementally. They can start speaking having in mind only a fragmentary idea of what they want to say, and while saying this they refine the contents underlying subsequent parts of the utterance. This capability imposes a number of constraints on the design of a syntactic processor…'

They concluded that, in human language generation, sentences are constructed incrementally as a series of chunks; phrase breaks are generated under local constraints, to fit immediate context rather than global sentence structure. Other psycholinguists studying human generation of specific language constructions reached the same conclusion. For example, Stallings *et al* (1998) investigated phrasal ordering constraints in sentence production, focussing on phrase length and verb disposition in Heavy-NP shift; they also concluded that sentences are constructed incrementally, with phrase breaks generated under local constraints:

> '...These findings point to the simultaneous activation of lexically derived syntactic representations and ordering options in sentence planning. A multiple constraints framework provides a means of reconciling the existence of competition among ordering options with incremental sentence construction…'

These findings within a broader cognitive framework suggest that a phrase break prediction model could be based on indicators in the immediately preceding words, and need notrely on sentence-level syntactic processing. Psycholinguists have also investigated how different classes of words contribute differently to cognitive processing in phrase construction. For example, Bell *et al* (2009) studied predictability effects on durations of content and function words in conversational English, and found that '...content and function words are accessed differently in phrase production...' and that there is '…a general mechanism that coordinates the pace of higher-level planning and the execution of the articulatory plan…' This suggests a broad content/function feature might be a useful indicator in phrase break prediction. This is discussed in some detail in section 3.2 of this thesis.

1.5. SEC as "gold standard"

We have already stated that the performance of a classifier is in part measured by the number of correct phrase breaks it recaptures in a sample of unseen text. The general procedure is to train the classifier on "gold standard" boundary annotated text from a speech corpus (the training set), and to hold in reserve a smaller section of text from the same source for testing. Although target boundary sites in the test set are available to the researcher for comparative evaluation, they are missing from test data presented to the classifier. Versions of SEC are often used as datasets for training/testing phrase break models (*e.g.* Taylor and Black, 1998; Read and Cox, 2007) because this corpus has already been marked up with boundary annotations by two corpus linguists: Gerry Knowles and Bryony Williams. These annotations are then regarded as a "gold standard" for developing and evaluating language models because they encapsulate human performance, that is the level or standard of performance *in terms of intelligibility and naturalness* that the model is designed to emulate. Tone unit boundary markers (|) and pauses (||) in SEC are *reactive* in that they represent annotators' perceptions of acoustic events in the speech signal, such as periods of silence, pre-boundary lengthening in word-final syllables, and presence/absence of coarticulatory effects (Dehe and Wichman, 2010), and *proactive* in the sense that they may also signify annotators anticipating or predicting the chunking strategy for a given sentence after exposure to a particular speaker or genre (Pickering *et al.*, 1996, p.65). Chapter 5 of this thesis includes further discussion of boundary annotations in SEC, especially the issues of inter-annotator agreement, and prosodic variance.

1.6. Ramifications of prosody

This thesis is concerned with prosodic phrasing and the prosodic-syntactic devices used (by speakers, listeners, readers and writers) to demarcate chunk boundaries, the beginnings and ends of meaningful units of thought. By way of illustration, we might consider possible chunking and highlighting strategies for the following sentence (Winograd, 1984).

> In the popular mythology the computer is a mathematics machine: it is designed to do numerical calculations. Yet it is really a language machine: its fundamental power lies in its ability to manipulate linguistic tokens – symbols to which meaning has been assigned.

1.6.1. Intuitive prosodic phrasing: Winograd extract

This exercise did not involve applying any explicit rules. Instead, the text was read aloud in an expressive way and choices about prominent words and resting places were tested over several readings to see if they could confidently be replicated. This is the result (Example 1.1), where chunking words (*i.e.* words preceding a prosodic phrase boundary) are given in *italics* and highlighted words in **bold**. Three instances of conflation also occur – on ***tokens*** and ***symbols*** and ***do***; the identification of the word ***do*** as a conflated item is considered particularly idiosyncractic.

Example 1.1

In the popular *mythology* the **computer** is a **mathematics** *machine*: it is designed to ***do*** numerical *calculations*. Yet it is **really** a **language** *machine*: its fundamental ***power*** lies in its ability to manipulate linguistic ***tokens*** – ***symbols*** to which **meaning** has been *assigned*.

We can use chunkers and highlighters to identify the most important lexical items, retaining the original linear order in which they appear: {mythology; computer; mathematics; machine; do; calculations; really; language; machine; power; tokens; symbols; meaning; assigned}. This list embodies a pretty powerful train of thought with some interesting word associations.

A problem arises, however, if we try to extract formal propositions from such complex sentence structure. Consider, for example, the first stand-alone section: *In the popular mythology, the computer is a mathematics machine*. It is easy to formulate the proposition: *the computer is a mathematics machine*. But this proposition is not strictly true, being qualified by the introductory prepositional phrase. Moreover, this prepositional phrase is key to introducing the contrast, reinforced by the stress, accenting and chunking, between a limited (popular) view of computers as calculators versus *reality* – computers as language machines. Finally, this prepositional phrase serves to associate computers with myth, with the human imagination, and yet this enriching aspect of meaning would be lost if we were to simply extract the proposition from the sentence and ignore the first chunk.

1.6.2. Prosodic versus syntactic phrase structure: Winograd extract

The nature of the relationship between prosody and syntax has been a continuing debate in the literature since the 1960s, with the intriguing paradox that prosodic phrasing both reflects syntactic constituency but is simpler, shallower and flatter than syntactic structure (Ladd, 1996). This is best illustrated by example.

Intuitively, we might break the following sentence up into 2 or 3 prosodic phrases (Example 1.2).

Example 1.2

The two-phrase version:
In the popular mythology || the computer is a mathematics machine ||

The three-phrase version:
In the popular mythology || the computer | is a mathematics machine ||

It does not matter which version we choose; what matters is that each chunk is meaningful in its own right and that boundaries are not aberrant occurrences as in this next version (Example 1.3).

Example 1.3

Nonsensical phrasing:
In the popular | mythology the | computer is a mathematics | machine |

A full parse of the above sentence shows that while prosodic structure is linear, syntactic dependencies create a multi-layer structure, traditionally represented as a parse tree (Figure.1.1).

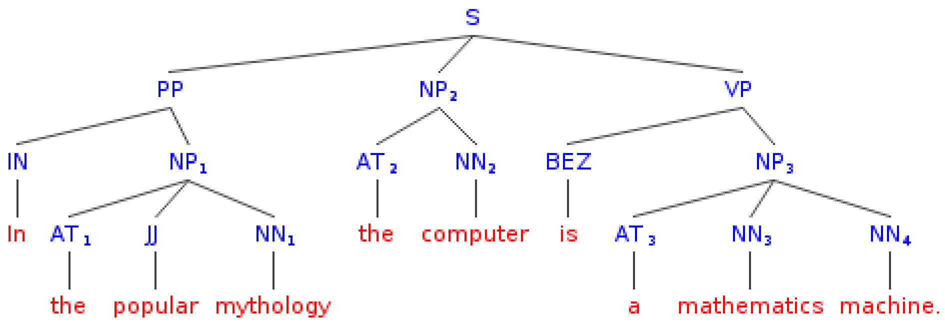


Figure. 1.1: Parse tree representation of example sentence.
http://www.ironcreek.net/phpsyntaxtree/

This tree was constructed from the following labeled bracket notation and uses Brown PoS tags to identify parts of speech at terminal nodes (Example 1.4).

Example 1.4

[S [PP [IN In] [NP [AT the] [JJ popular] [NN mythology]]**]** **[NP** [AT the] [NN computer]**]** **[VP** [BEZ is] [NP [AT a] [NN mathematics] [NN machine.]]**]]**

The example suggests that prosodic phrase breaks equate to the nodes marked in **bold** in this bracketed notation and that they occur between large syntactic units {NP, VB, PP, ADJP, ADVP}. This intuition is included in the selection of features used in a CART (Classification and Regression Tree) model for automatic phrase break prediction (Wang and Hirschberg, 1991) which reports a 90.8% success rate in the detection of prosodic boundaries.

1.7. Structure of thesis document

This thesis is structured as follows. Chapter 2 reviews speech corpora and prosodic annotation schemes, and Chapter 3 reviews deterministic and stochastic approaches to phrase break prediction, plus customary evaluation metrics and feature sets. Chapter 4 reports on early rule-based experimental work with a shallow or chunk parser. Outputs from same prompt closer inspection of corpus annotation and give rise to motivational insights about prosodic variance (Chapter 5) which then inform resource creation (Chapter 6). Prosodic resources comprise: ProPOSEL, a ***pro***sody and ***p***art-***o***f-***s***peech ***E***nglish ***l***exicon, supported by a comprehensive software tutorial (Appendix 2), and the ProPOSEC dataset (§8.10). Significance testing finds a high degree of correlation between gold standard boundary annotations and certain sound patterns (*i.e.* complex vowels) in English in different genres: seventeenth century English verse (Chapter 7), plus read speech (*i.e.* a lecture) and spontaneous speech from the twentieth-century (Chapter 8). Linguistic data arrays in this spontaneous speech dataset (*i.e.* ProPOSEC) are then re-conceptualised as training instances for supervised machine learning experiments (Chapter 9) and the final summary, conclusions, and ideas for further work appear in Chapter 10.

Chapter 2
Speech Corpora and Prosodic Annotation Schemes

2.1. English speech corpora used in this thesis

The main speech corpora used for experimental work in this thesis are the Lancaster/IBM Spoken English Corpus (Taylor and Knowles, 1988) and a more recent version of this dataset: the Aix-MARSEC corpus project (Auran *et al.*, 2004). These are corpora of larger texts, with content beyond words and phrases, as opposed to speech corpora such as TIMIT (§2.5). The latter has been used to train acoustic models for Automatic Speech Recognition, while SEC can be used to develop better rules for Text-to-Speech Synthesis (Knowles, 1996a). The Spoken English Corpus and Aix-MARSEC are outlined in this and the subsequent section on prosodic annotation schemes, along with other relevant speech corpora.

2.2. The Spoken English Corpus

The version of the Spoken English Corpus (SEC) used in this thesis is available from NLP resources in the School of Computing at Leeds University and vertically aligns each word in the corpus with its part-of-speech classification from the Lancaster-Oslo-Bergen (LOB) tagset (Johansson *et al.*, 1986). The original SEC is a corpus of some 52,000 words of contemporary British English speech collected at the University of Lancaster, and transcribed orthographically and prosodically, plus annotated grammatically via the CLAWS (Constituent Likelihood Automatic Word-tagging System) tagger and CLAWS1 (*i.e.* LOB) tagset (UCREL, 2010). High quality recordings of speech samples from 53 different speakers and produced by IBM UK are also included as part of the corpus; most of these samples are taken from BBC radio. There are 11 speech categories in all: {A: Commentary; B: News broadcasts; C: Lecture type 1; D: Lecture type 2; E: Religious broadcast; F: Magazine style reporting; G: Fiction; H: Poetry; J: Dialogue; K: Propaganda; M: Miscellaneous}. Table 2.1 gives additional information for sections used in this thesis (*i.e.* Sections A and C) .

Subsection	**Source**	**Speaker(s)**	**Date**
A01	In Perspective	Rosemary Hartill	24.11.1984

A02 – A06	From our own Correspondent	Gerald Butt; Jon Silverman; John Carlin; James Morgan; David Smeeton	24.11.1984
A07 – A12	News	Laurie Margolis; Keith Graves; Graham Leach; Alan MacDonald; Peter Ruff; Jim Biddulph	22.06.1985
C01	The Reith Lectures (III)	David Henderson	20.11.1985

Table 2.1: Genre and speaker identification in a sample from SEC

2.3. The Aix-MARSEC Corpus Project

The Aix-MARSEC corpus of Spoken British English is described as a freely available, collaborative and evolving database where the plan is to incorporate further contributions by referenced users. It originates from the Spoken English Corpus, comprising over 5 hours of BBC radio recordings from the 1980s, and its machine readable counterpart: MARSEC (Roach *et al.*, 1993). The original prosodic annotations have been augmented by multi-level annotation tiers from the Aix-MARSEC project and these are discussed in more detail in Section 2.8.

2.3.1. The Aix-MARSEC toolset

Aix-MARSEC is both a toolset and a database. The former includes tools specifically designed for use on the database – multiplatform Praat and Perl scripts and reference files – and a general purpose prosody editor (PROZED for short) incorporating the MOMEL-INTSINT algorithms for reproducing the prosodic characteristics of utterances (Hirst, 2000a).

2.3.2. The Aix-MARSEC database

The Aix-MARSEC database comprises radio recordings of fifty-three different speakers, in eleven different speech styles, with prosodic annotations from two experts; plus a further nine levels of prosodic annotation presented as separate tiers in Praat TextGrids. Two further levels of syntactic annotation were originally planned for syntactic annotation and a property grammar system.

2.4. Hand-labelled speech corpora

The International Computer Archive of Modern and Medieval English (ICAME) website lists several speech corpora where the recordings have been transcribed and prosodically annotated according to the British tone sequence model (§1.4), where intonation is described as a succession of pitch movements – falling or rising tones, for example. However, these corpora have been hand-labelled, prosodic marks used are corpus-specific and transcriptions are not intended for automatic processing. They are included here to highlight some of the important characteristics of prosody.

The London-Lund corpus (Greenbaum and Svartvik, 1990) represents some thirty years' work and contains a hundred spoken texts – both spontaneous and composed monologues and dialogues – with detailed prosodic information identifying the following key features: tone unit boundaries; pauses of varying lengths; location of the nucleus or most prominent peak; direction of nuclear pitch accent; "boosters," (*i.e.* resetting of pitch level between one tone unit and another); varying degrees of stress, loudness and tempo; changes in voice quality; and paralinguistic features. This is a complex body of information.

An alternative approach has been adopted in COLT, the Bergen Corpus of London Teenage Language (University of Bergen, 1993), which contains some half a million words of spontaneous and lively speech by 13 to 17 year olds from five very different districts in London. Parts of this corpus have been transcribed prosodically, with a practical and straightforward set of labels comprising: `[#]` for tone unit boundary; `[-]` for level tone; use of **bold type** to mark nuclei; and the set { `\, /, \/, /\` } to designate direction of pitch accent: falling, rising, fall-rise and rise-fall. This annotation set resembles the scheme used in the Lancaster/IBM Spoken English Corpus discussed in Section 2.8.

2.5. LDC corpus holdings

The Linguistic Data Consortium (LDC) holds numerous speech corpora, each developed for a specific purpose and application and classified according to type as a shared resource for the international research community. The following are notable catalogue inclusions used in studies.

The target application for the Boston University Radio Speech Corpus (Ostendorf *et al.*, 1996) was TTS, with an emphasis on the generation of prosodic patterns. The corpus consists of over seven hours of professionally read radio news by four male and three female announcers recorded over a two year period. An interesting feature of this corpus is the additional laboratory recordings of the same speakers reading the same news items in two contrasting styles: their normal speaking voice and their radio broadcast speech style. Annotations include orthographic transcription, phonetic alignments, part-of-speech tags and prosodic markers, including pauses, but only for a subset of the corpus.

TIMIT, the Acoustic-Phonetic Continuous Speech Corpus (Garofolo *et al.*, 1993), is a corpus of read speech designed to provide data for Automatic Speech Recognition (ASR). It contains broadband recordings of 630 speakers of 8 major dialects of American English, each reading 10 phonetically rich sentences and thus has immediate potential for the statistical analysis of intonational variation between native English speakers. The TIMIT corpus has time-aligned orthographic, phonetic and word transcriptions which have been hand-verified, plus a 16-bit, 16kHz speech waveform file for each utterance and specified test and training subsets.

The Switchboard Telephone Speech Corpus (Godfrey and Holliman, 1997) was also gathered for ASR. It is a collection of some 2400 telephone dialogues involving 543 male and female speakers from across the US on a range of topics and hence lends itself to research applications such as discourse annotation and the annotation of speech acts. An interesting feature of this corpus is that speaker attributes have been listed.

2.6. The British National Corpus

The BNC or British National Corpus (Burnard, 2000) has been reissued as a third edition in XML format so that it can be used with state-of-the-art natural language processing tools and resources. The corpus was completed in 1994 and contains some 100 million words of modern British English. The spoken part constitutes 10% of the whole and is made up of recordings and transcriptions of spontaneous speech by members of the public from 38 different UK locations, plus context-specific material from educational, business, public and leisure events. This corpus is not prosodically annotated, however.

2.7. Other speech corpora

The SPOT corpus (Speer *et al.*, 2000) was set up as a cooperative game task involving 16 pairs of male speakers of American English to investigate how prosody can be used to resolve syntactic ambiguity. Findings showed a match between prosodic and syntactic phrasing in the controlled grammatical context under investigation – transitive and intransitive sentences – but also some variety in tone sequences used by participants even in tightly scripted speech.

The IViE, or International Variation in English, corpus project (Grabe *et al.*, 2001) constitutes thirty-six hours of speech recordings from sixteen-year-olds from nine urban dialects in the UK. A small section of this corpus has been prosodically transcribed.

The OXIGEN project, or ***Ox***ford ***I***ntonation ***Gen***erator, (Grabe *et al.*, 2003) follows on from IViE and aims to further the investigation of demographic variation in intonation patterns in the UK. OXIGEN is intended to provide a statistical computational model of intonation in English for TTS and ASR which takes account of influences such as region, gender and individual speaking style.

2.8. Prosodic annotation in SEC and Aix-MARSEC

The Spoken English Corpus was annotated by Knowles and Williams according to the conventions of the British School (Grabe, 2001) where pitch accent types are described as pitch movements. There are, in effect, *fifteen* prosodic marks used in SEC-MARSEC representing the full range of accent types for British English; plus two types of boundary corresponding to break indices 3 and 4 in the ToBI system (§2.9); a mark signifying hesitation tone unit boundary; a mark for stressed as opposed to accented syllables; and finally two reset labels for higher or lower than predictable pitch. The latter are also used in the INTSINT coding system in Aix-MARSEC.

Prosodic annotation in the Aix-MARSEC corpus presents the following information about an utterance in a nine-layer Praat (Boersma and Weenink, 2009) TextGrid. Words in the orthographic tier are broken down into individual phonemes and syllables in separate tiers, with a further tier revealing syllable structure. All this information is aligned. Furthermore, there is a series of separate tiers for suprasegmental components arranged in a hierarchy of stress feet, narrow rhythm units and ancruses and finally intonation units – and again, all this information is aligned. Yet another tier represents INTSINT coding of intonation at the surface phonological level. The MOMEL-INTSINT algorithms use a sequence of target points derived from the acoustic signal to enable automatic modelling of fundamental frequency curves at the level of phonetic representation; and automatic coding of intonation using a finite set of symbols: {T, M, B, H, L, S, U, D}. This set of symbols {Top, Middle, Bottom, Higher, Lower, Same, Upstepped, Downstepped} does not constitute a fixed inventory of labels (for pitch accents and boundary tones, for example) as used in Autosegmental-Metrical annotation schemes such as ToBI, ToDI and IViE (§2.9) but instead uniquely codes the intonation of an utterance as a pattern of absolute and relative tones. The final tier gives fundamental frequency values. Table 2.2 cross-references prosodic annotation marks used in SEC with those used in Aix-MARSEC.

<table>
<tr><th>SEC</th><th>Aix-MARSEC</th><th>Signification</th></tr>
<tr><td>_</td><td>_</td><td>Low level</td></tr>
<tr><td>|</td><td>|</td><td>Minor tone unit boundary</td></tr>
<tr><td>||</td><td>||</td><td>Major tone unit boundary</td></tr>
<tr><td>¯</td><td>~</td><td>High level</td></tr>
<tr><td>↓</td><td><</td><td>Lower than predictable pitch</td></tr>
<tr><td>↑</td><td>></td><td>Higher than predictable pitch</td></tr>
<tr><td>⇑</td><td>none</td><td>Hesitation tone unit boundary</td></tr>
<tr><td>˅</td><td>/'</td><td>High rise-fall</td></tr>
<tr><td>˄</td><td>'/</td><td>High fall-rise</td></tr>
<tr><td>\</td><td>\</td><td>High fall</td></tr>
</table>

/	/	High rise
\	'	Low fall
/	,	Low rise
˅	,\	Low rise-fall (not used)
˄	\,	Low fall-rise
•	*	Stressed but unaccented

Table 2.2: Prosodic marks cross-referenced in SEC and Aix-MARSEC, as presented in Taylor and Knowles (1988) and Auran *et al.* (2004) respectively

2.9. Autosegmental-Metrical (and other) prosodic annotation schemes

ToBI (***To***nes and ***B***reak ***I***ndices) - is a machine-readable prosodic annotation scheme where transcriptions aim to capture features in the acoustic signal and a speaker's phonological choices. It has been widely used in studies because of its descriptive inventory of tones facilitating cross-linguistic comparison. ToBI has been through several revisions since its first appearance; the final set of four transcription tiers (Pitrelli, Beckmann and Hirschberg, 1994) includes a break index tier and a tone tier. In the former, values are assigned to disjuncture between words and intonational phrases, introducing a notional distinction between an intermediate phrase (Break Index 3) and a full intonation phrase (Break Index 4) and leading to theories outlining a hierarchy of prosodic constituents. A similar distinction between shorter and longer pauses, often corresponding to commas and full-stops, is made in other prosodic annotation schemes. The ToBI tone tier includes labels such as the high/low phrase accent (H-, L-); the phrase-initial boundary tone (%H, %L); and the final boundary tone (H%, L%). Table 2.3 illustrates ToBI annotations for pitch accents and boundaries, juxtaposed with boundary annotations from SEC.

Orthographic Tier	**Will**	**you**	**have**	**marmalade,**	**or**	**jam?**
Tone Tier				L* H-		L* H-H%
Break Index Tier	1	1	1	3	1	4
SEC scheme				|		||

Table 2.3: Illustration of American (ToBI) prosodic annotations for sample sentence, including comparison with boundary annotations in SEC

ToDI is a two-tier annotation scheme now in its second edition (Gussenhoven, Rietveld, Kerkhoff and Terken, 2003) which uses a modified inventory of ToBI-like tones to label speakers' choices at significant points mapped to the orthographic tier. It was devised for the Dutch language but is applicable to British English, particularly since the inventory includes in its terminology all six pitch accent types mentioned in Section 1.4.

The IViE (***I***ntonational ***V***ariation ***i***n ***E***nglish) notation system is a further variant of the Autosegmental-Metrical approach inaugurated by ToBI and was devised to annotate a corpus of the same name with comparative recordings from nine different locations in the British Isles (Grabe *et al.*, 2001). IViE was released in 2001 and is distinctive in its incorporation of two new annotation tiers: a prominence tier and a phonetic tier. The prominence tier labels each peak in the frequency contour as `P`, and maps this symbol to the corresponding syllable in the orthographic tier. Furthermore, the phonetic tier introduces the concept of an implementation domain spanning consecutive segments or eventful blocks of the frequency contour and labels unaccented syllables either side of significant prominences (the "P"s) as high/low pitch or mid-range.

The Tilt model (Taylor, 2000) is described as an event detector which subsequently generates a synthesized intonation contour from the acoustic signal – a phonetic model, therefore. Synthesized contours can then be compared to real ones. Significant events for this model were initially defined as pitch accents and rising boundaries, labeled `[a]` and `[b]` respectively, plus combination events – `[ab]` – where accent and boundary are realised as a single pitch movement. Presumably, the default boundary type – a fall – was not interpreted as an event because it marks the end of the utterance or segmental stream input. The results of automatic event detection trials were compared with human transcriptions of the same input, where annotators were given a fuller set of labels in keeping with traditional annotation schemes. This label set included `[sil]` for periods of silence; `[l]` for level accents; `[m]` for stressed but unaccented syllables; and labels for differentiating rising, level and falling accents in combination events. It was found that augmenting the model with a fuller set of labels gave better performance.

2.10. Phonetic transcription schemes used in this thesis

The Prosody and PoS English Lexicon (ProPOSEL) described in Chapter 5 specifies phonetic transcriptions for each word form entry via two different character sets: SAM-PA and DISC. ARPAbet transcriptions have also been used during lexicon build to generate some of the lexical stress patterns (§6.3.1), the ARPAbet being a phonetic transcription scheme using ASCII symbols and specifically designed for American English.

SAM-PA is a standard computer-readable phonetic character set and is used in ProPOSEL for transcriptions derived from CUVPlus (Pedler and Mitton, 2003). As with the International Phonetic Alphabet (IPA), SAM-PA uses 2 characters to represent affricates (*i.e.* /tS/ and /dZ/ in *chin* and *gin*) and diphthongs (*e.g.* /eI/ and /aI/ in *day* and *night*). In contrast, the DISC (***Di***stinct ***S***ingle ***C***haracters) set implements an unambiguous, one-to-one mapping of character to segment in the sound systems of Dutch, English and German and is recommended for computer processing tasks (Burnage, 1990). Table 2.4 shows phonetic transcriptions in ProPOSEL for the homograph *attribute* and the noun *foundation*, which contains 2 diphthongs (*cf.* Table 7.5); SAM-PA transcriptions are stressed and DISC transcriptions are stressed and syllabified.

Word	**C5 PoS tag**	**SAM-PA**	**DISC (stressed & syllabified)**	**DISC (stress weightings assigned to each syllable)**
attribute	NN1	'&trIbjut	'{-trI-bjut	'{:1 trI:0 bjut:0
attribute	VVB	@'trIbjut	@-'trI-bjut	@:0 'trI:1 bjut:0
attribute	VVI	@'trIbjut	@-'trI-bjut	@:0 'trI:1 bjut:0
foundation	NN1	faUn'deISn	f6n-'d1-SH	f6n:0 'd1:1 SH:0

Table 2.4: Example entries from ProPOSEL for word; part-of-speech; and SAM-PA and DISC phonetic transcriptions

2.11. Syntactic annotation schemes relevant to this thesis

Generally, in corpus compilation, part-of-speech (PoS) tags constitute the first level of linguistic enrichment for a text (Atwell, 2008), providing much more discriminating syntactic information per word than is found in dictionaries. One motivation for ProPOSEL (*cf.* Chapter 6) was to construct a lexicon for linkage with a range of speech corpora, necessitating, therefore, inclusion of more than one syntactic annotation scheme – four in all. ProPOSEL inherited the C5 tagset (Leech and Smith, 2000) from CUVPlus; this is a fairly sparse tagset (just over 60 tags) designed specifically for handling large quantities of data, as in the British National Corpus (Burnard, 2000). Inclusion of LOB or the Lancaster-Oslo-Bergen tagset (Johansson, 1986) was essential for cross-referencing syntactic information from SEC with Aix-MARSEC, since to date, the latter is not syntactically annotated. In addition, there are fields in ProPOSEL for Penn Treebank tags – as used in TIMIT and the Switchboard Telephone Speech Corpus, for example – and for C7, since this is UCREL's current standard tagset (UCREL, 2010). Of the 4 tagsets used in ProPOSEL, C7 is the most fine-grained and Penn the least. The scheme for mapping C5 to Penn, LOB and C7 is discussed in Chapter 6.5.2 and full details are given in Appendix 2: ProPOSEL's software tools.

Chapter 3
Prosodic Phrase Break Prediction: Methods, Metrics and Feature Sets

3.1. Overview of task

Techniques for automated prediction of prosodic phrase boundaries in text, typically for Text-to-Speech Synthesis (TTS) applications, can be deterministic or probabilistic. In either case, the problem of phrase break prediction is treated as a classification task and outputs from the model, as in other Natural Language Processing (NLP) applications such as part-of-speech (PoS) tagging, are evaluated against a human-labelled 'gold standard' corpus (Jurafsky and Martin, 2000 p.308), also known as a 'reference dataset' in the speech research community. For prosody, this gold standard is a test set where original transcriptions of recorded speech in the speech corpus include prosodic annotations by experts. Annotation systems commonly used for phrase break prediction are ToBI - Tones and Break Indices (Beckman & Ayers, 1997) - where the break index tier distinguishes 5 levels of juncture between words on a scale of 0 - 4, and the British system exemplified in SEC - the Spoken English Corpus (Taylor and Knowles, 1988) - which identifies 3 levels: no boundary, minor phrase boundary, major intonational phrase (IP) boundary. Minor and major boundaries are assigned the pipe symbols: **|** and **||** respectively, and map to break indices 3 and 4 in ToBI. In Roach (2000), these same symbols denote *tone unit boundary* **|** and *pause* **||**.

3.2. Rule-based methods

A standard rule-based method commonly used in TTS is to employ some form of 'chink-chunk' algorithm which inserts a boundary after punctuation and whenever the input string matches the sequence: open-class or content word (chunk) immediately followed by closed-class or function word (chink), based on the principle that chinks initiate new prosodic phrases. Bell Labs speech synthesizer has used this kind of rule to identify low-level phrasal units or f-groups (Abney, 1994); and a similar notion of f-groups or function word *constraints* has been used to cluster parts-of-speech sandwiched between two function words (Elliott, 2003). Variants of the chink-chunk algorithm may seek to shuffle parts-of-speech (PoS) between open and closed-class groupings; the chink-chunk algorithm proper (Liberman and Church, 1992) treats tensed verb forms as chinks and object pronouns as chunks for more natural phrasing. The challenge for this algorithm, which tends to over-generate boundaries, is to discover rules for merging f-groups into more complex units. Pertinent *syntactic* principles have been posited by Knowles (1996a) and Croft (1995) for mapping grammatical units onto intonation units. The former recommends lightening dense intonation by looking at *local*, rather than global, relationships between grammatical units: in effect by ranking successive f-group blocks (Knowles, 1996a, pp. 150; 153). For the latter, the most important constraint is avoidance of parallel structures, as distinct from nested structures, within single intonation units. This thesis is interested in *phonetic* and *phonological* principles influencing such a mapping.

A more recent alternative rule-based method is described by Atterer (2002) and Atterer and Klein (2002); their model builds a hierarchical prosodic structure via a two-step process which uses the CASS chunk parser (Abney, 1991) to identify φ-phrases (f-groups) and then 'bundles' these minor phrases into intonational phrases. The algorithm uses a variable threshold figure (default setting 13) to limit the number of syllables in an intonational phrase if there is no intervening punctuation. This approach is reminiscent of Miller (1956) and the argument that humans have a short term or 'immediate' memory span manifest in our tendency to process information in a fixed *number* of chunks; (this appears to be a specific use of the word 'chunks' as *countable units*). However, while Atterer and Klein here define a chunk as a sequence of words which amounts to *no more than 13 syllables*, Miller seems less definite about size of chunk. For example, he refers to a set of experiments by Hayes (1952), where data consisted of 1000 monosyllabic words (not connected speech, therefore), and where human subjects were determined to have an immediate memory span of five words and 15 phonemes, "...since each word had *about* 3 phonemes in it..." [my italics]. This thesis suggests that a more appropriate predictor for intonational phrases in English would be beats (*i.e.* syllables which carry primary stress) rather than phonemes, syllables *per se* and individual words (*cf.* 6.6.3; 9.5.1).

3.3. Statistical methods

The leading study in the use of statistical methods for phrase break prediction is Taylor and Black's Markov model (1998), trained and tested on MARSEC, the Machine Readable Spoken English Corpus (Roach *et al*, 1993) and used in Edinburgh's Festival speech synthesis system (Black *et al.*, 1999; Black, 2000). The training data for this supervised learning model is 'text' represented by a sequence of PoS tags which include boundary tags. The model is structured such that states represent types of break - the desired classification outputs of break or non-break - and transitions represent likelihoods of phrase break sequences occurring. The model thus 'learns' the classification task by integrating two sets of information: the probability of a PoS sequence, given juncture type, and the probability of a particular sequence of juncture types occurring. This extensive study actually goes on to compare the performance of both probabilistic *and* deterministic language models over six experimental settings, with a best score of 79% breaks-correct achieved with a higher order n-gram model and a more streamlined tagset obtained by post-mapping the output of the PoS-tagger onto a smaller tagset of 23.

Busser *et al*. (2001) compare the effectiveness of a Memory-Based Learning (MBL) approach to predicting phrase breaks in MARSEC to Taylor and Black's ('gold standard') use of HMMs for the same purpose. MBL is a supervised-learning approach where classification of data is made on the basis of maximum similarity to items in memory. In this study, the set of feature values descriptive of phrase break contexts, and used as input to train the classifier is: the orthographic form of the word in question; its PoS tag; its CFP-value (status as content word, function word or punctuation mark); and an expanded tag which gives the word itself if it is a function word and the PoS tag otherwise. A fixed-width feature vector of two words both to the left and right of the focus position in question supplies the context from which to extrapolate the 'minority' class 1 (break) or more frequent class 0 (non-break i.e. ordinary juncture). The study involves converting Taylor and Black's results over six experiments to the MBL metrics of precision, recall and F-score (see the discussion on performance measures in section 3) for the purposes of comparison and then experimenting with further optimization of these metrics, creating a different mix of information in the feature vectors via leave-one-out experiments and cross-validating against the training set. Busser *et al*. report an improvement on the best HMM result for recall with a simple MBL algorithm which takes a limited context of one PoS to the left and right of the focus position and assigns equal weighting to each of these positions.

Taylor and Black's use of a reduced tagset in their framework for assigning phrase breaks from PoS information has been taken forward in a recent study by Read and Cox (2004). This presents a best first search algorithm (suitable for any tagset) for exploring and determining groupings of PoS tags that will eventually constitute a reduced, optimal tagset for phrase break prediction. Read and Cox use what they term a *flattened* prosodic phrase hierarchy classification of break/non-break on datasets from the Boston Radio News Corpus (Ostendorf *et al.*, 1995) and MARSEC, and to evaluate their phrase break prediction model, use Taylor and Black's *junctures correct* measure, that is the percentage of non-breaks correctly predicted.

The statistical modelling technique known as CART (a Classification and Regression Tree) is used by Wang and Hirschberg (1991) to predict prosodic phrase boundaries from features that can be automatically generated from text. 'Learning' for this decision tree method includes training the splitting rules at each decision point in the tree to select the feature/value split which minimises prediction error rate in the training set. In this study, such features include: length of utterance in seconds and words; position of potential boundary site and distance from beginning and end of utterance; and syntactic constituents adjacent to the boundary site. An important additional feature used to compare the performance of the original model to an enhanced model which incorporates hand-labelled transcriptions in the data set (298 sentences of air travel information from DARPA, 1990) is accent status of <*wi*>, where <*wi* , *wj*> represents words either side of the boundary site. The best performing variable set included information from prosodic annotations of pitch accent and prior boundary location, giving a success rate of 90% boundaries correct and a streamlined tree with only 5 decision points.

A related and more recent study (Koehn *et al.*, 2000) builds on an augmented version of the above feature set (Hirschberg and Prieto, 1996) by adding syntactic information from a high accuracy syntactic parser. The '1996' feature set consists of the following: a 4-word PoS window and a 2-word accent window; the total number of words and syllables in the utterance; word distance from start and finish of the utterance in words, syllables and stressed syllables; distance from last punctuation mark and what punctuation, if any, follows the word; position of word in relation to, or within, a noun phrase; and finally, size and distance of word from start of noun phrase. The ‘2000’ feature set builds on the intuition that prosodic phrase breaks occur between large syntactic units {NP, VB, PP, ADJP, ADVP} and incorporates binary flags indicating which words initiate a major phrase or a sub-clause. The study reports a 90.8% prediction rate for boundary detection which is cross-validated using other machine learning algorithms: a boosting algorithm, a rule learner, a boosted decision tree classifier and an alternating decision tree method.

3.4. Evaluation metrics used in studies

The previous section briefly discusses a range of machine learning methods applied in prosodic phrase break prediction. The evaluation metrics used in studies seem to fall into one of two groups, however. The first group (see Wang and Hirschberg, 1991; Atterer, 2002; Read and Cox, 2004) select from the set of *accuracy* and *error* measures discussed in Taylor and Black (1998) and presented in Tables 3.1 and 3.2.

Taylor and Black argue that *breaks-correct* is a better measure of algorithmic performance than *junctures-correct* (Table 3.1) because the latter includes *non-breaks* in the calculation and these are always more numerous.

% breaks-correct (true positives)	breaks correctly predicted/ total number of breaks in test set	x 100
% non-breaks correct (true negatives)	non-breaks correctly predicted/ total number of non-breaks in test set	
% junctures-correct	(breaks + non-breaks) correctly predicted/ total number of junctures in test set	

Table 3.1: Accuracy measures for phrase break prediction

% insertion errors (1) (*false positives*)	breaks retrieved by model / total number of breaks in test set	x 100
% insertion errors (2)	breaks retrieved by model / total number of junctures in test set	
% deletion errors (*false negatives*)	breaks missed by model / total number of breaks in test set	

Table 3.2: Error measures for phrase break prediction

The second group of evaluation metrics employed in statistical NLP (and for phrase break prediction see the aforementioned: Koehn *et al.*, 2000; Busser *et al.*, 2001; Atterer and Klein, 2002) are taken from the field of Information Retrieval and are known as *precision* and *recall*. The latter corresponds exactly to the *breaks-correct* measure, while the former equates to *positive predictive value*: in this case, the proportion of correct (relevant) predictions out of all the predictions made. In practice it is usual to combine precision and recall into a single overall performance measure or F-score which tends to maximise *true positives* (Manning and Schütze, 1999) - in this case *breaks-correct*. Table 3.3 shows how *precision*, *recall* and *F-score* are interpreted for the task of phrase break prediction.

Precision	breaks correctly predicted / number of breaks retrieved	x 100
Recall	breaks correctly predicted / total number of breaks in test set	
F-score	2 * precision * recall / precision + recall	

Table 3.3: Information Retrieval measures used in phrase break prediction

3.5. The elusive gold standard for prosodic phrasing

Phrase break prediction models are evaluated in terms of their ability to match boundary annotations in the test corpus. However, the long-term view is that the model will be able to generate intelligible and natural prosodic phrasing for *any* input text. It is hoped the model will have learnt the classification task well enough to make generalisations from the gold standard to the new domain. If it hasn't, it runs the risk of imposing a prosody template (one speaker, one realisation, one moment in time) on unsuspecting text. Some models over-predict; but how many of their false insertions or false positives are nevertheless valid in terms of performance structure? How many missed boundaries or false negatives in a given model are significant omissions? Perhaps the only way to answer these rhetorical questions would be to re-evaluate output predictions from the language model, assuming this model has already satisfied performance targets in terms of the conventional accuracy measure. Evaluation would then necessitate text marked up with all plausible boundaries, and entail subjective human judgements as to intelligibility and naturalness of posited boundary sites. Such data is not available on a sufficiently large scale to obtain significant results, and since we only have access to one prosodic variant per utterance, is beyond the scope of this thesis.

3.6. A closer look at features used in phrase break prediction

So far, the discussion has focused on techniques and evaluation metrics used in phrase break prediction, plus the inherent problem of prosodic variance: more than one natural and intelligible phrasing (*i.e.* more than one gold standard) exists for most sentences; and models trained on one corpus may not generalise to other domains. This is confirmed in a recent study (Read and Cox, 2007) where the same feature set achieved different results on two different speech corpora: (i) an f-score of 81.6% on MARSEC, versus (ii) an f-score of 77.9% on the Boston Radio Speech Corpus, indicating that '…choice of features is sensitive to the material used…'

This section revisits features and feature sets typically used in phrase break prediction.

3.6.1. Syntactic features

Syntactic features are integral to phrase break prediction because of the overlap between syntactic and prosodic phrasing. Table 3.4 gives a possible parse and a simplified view of human consensus (✓) on the best place to pause in this complex sentence, one of fourteen used in landmark psycholinguistic studies (Grosjean *et al.*, 1979; Gee and Grosjean, 1983); rules for English grammar normally require a comma in this position.

Subordinate clause							**Main clause**				
After	the	cold	winter	of	that	year	most	people	were	totally	fed-up
						✓					
[S [S [PP After [NP the cold winter]**]** **[PP** of [NP that year]**]]** **[S [NP** most people**]** **[VP** were [ADJP [ADVP totally] fed-up]**]]]**											

Table 3.4: The most likely within-sentence phrase break corresponds to a major syntactic boundary in this sentence, where the parsing strategy (i.e. bracketing) is given by Link grammar

3.6.2. CFP tags

Besides punctuation (which is really a *text-based* feature), the least sensitive and most transferable *syntactic* feature for predicting phrase breaks is content-function word status. For our model sentence, content-function word boundaries identify four phrasal units corresponding to major syntactic groupings defined by the Link parser (Sleator and Temperley, 1991) as shown in Table 3.5.

PP	**PP**	**NP**	**VP**
After the cold winter	of that year	most people	were totally fed-up.

Table 3.5: Function-word groups captured by a standard CFP algorithm here match syntactic units from the Link parser

3.6.3. PoS tags

Festival's speech synthesis system requires more discrete syntactic information in the form of PoS tags. PoS tagging is an automated process with accuracy rates as high as 96-97%; our example sentence has been assigned the following set of C5 tags via Lancaster's free online CLAWS trial service (UCREL, 2010) as shown in Table 3.6.

After	the	cold	winter	of	that	year	most	people	were	totally	fed-up
CJS	AT0	AJ0	NN1	PRF	DT0	NN1	DT0	NN0	VBD	AV0	AJ0

Table 3.6: Words are classified via C5 PoS tags by the CLAWS free trial PoS-tagging service.

3.6.4. Parse features

Building on the intuition that phrase breaks occur between major syntactic units {NP; VP; PP; ADJP; ADVP}, Koehn *et al.*, (2000) augment a sophisticated feature set with binary flags indicating whether or not the token initiates a major phrase or sub-clause. Their impressive prediction rate of 90.8% for boundary detection is partly accounted for by their incorporation of a feature derived from hand-labelled transcriptions: *i.e.* accent status of words adjacent to the boundary site; whereas the aim is to predict prosodic events like phrase breaks and accents automatically.

3.6.5. Text-based features

Taylor and Black, and more recently (Ingulfsen *et al.*, 2005), have demonstrated that punctuation is the single most important source of information for phrase break classification, finding approximately 50% of all breaks. Other features automatically generated from text and transcribed speech, and used to supplement syntactic features, include: word counts denoting length of utterance and distance of potential boundary site from start and end of sentence (Wang and Hirschberg, 1991); total number of words and syllables, plus distance from start and finish of utterance in words, syllables and stressed syllables, plus distance of potential boundary site from last punctuation mark (Hirschberg and Prieto, 1996; Koehn *et al.*, 2000).

3.6.6. Combined feature sets

Recent work (Ingulfsen *et al.*, 2005) revisits syntactic features to determine the effectiveness of deep versus shallow linguistic representations for phrase break prediction. They find that a shallow representation (CXPoS) which provides different levels of granularity via (i) CFP tags; (ii) an expanded tag giving the word itself if it is a function word and the PoS tag otherwise; and (iii) PoS trigrams of two tags before and one after the boundary site (Taylor and Black, 1998), is as effective as the best performing deep representation. The latter augments the PoS trigrams with novel Link grammar parse features extracted as outputs from the Collins parser, where *links* are labelled arcs showing syntactic coupling between words, as illustrated in Figure 3.1.

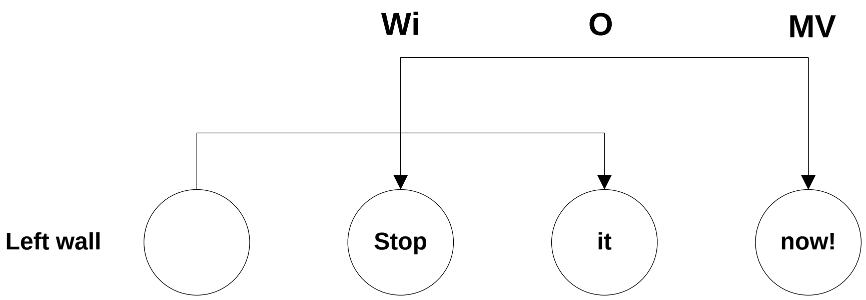


Fig.3.1: Labelled arcs for the string *Stop it now!* show syntactic coupling between *Stop it* (imperative {Wi} and object {O}) and *Stop now* (imperative {Wi} and modifier {MV}) but no link between *it* and *now*, permitting a possible phrase break in this position.

The best performing model in this study uses Link features to complement CXPoS and achieves a good balance between the Information Retrieval metrics of precision (64.7%) and recall (64.4%).

Read and Cox (2007) use various techniques and syntax-based feature sets to model the prosody-syntax interface. As with Ingulfsen *et al.*, their best classifier combines deep and shallow features: long-range parse features derived from the Collins parser and expressed as the size of the biggest phrase ending in the current word, plus a binary flag indicating whether the phrase belongs to the set {NP; VP; PP; ADJP; ADVP}; and n-gram probabilities for both classes derived from a localised PoS window (trigrams), where a reduced tag set of 7-8 PoS is used (Read and Cox, 2005).

3.7. Incorporating non-traditional features

Ananthakrishnan and Narayanan (2008) adopt a novel approach to automatically annotating speech corpora by attempting to integrate the prediction of accents and boundaries based on combined feature streams (acoustic, lexical and syntactic), their dataset being the Boston University Radio News Corpus. Their study associates prosodic events with specific syllables and does so on the basis that syllables are the smallest linguistic units from which prosodic phenomena can be detected.

Syllable *counts* have previously been implemented in syntax-based phrase break models for English to regulate the number of syllables in any one intonational phrase (Atterer, 2002; Atterer and Klein, 2002); and as a distance metric for encoding global information in the sentence (*i.e.* distance in syllables from the last break), which is then used to condition prior probabilities for breaks and non-breaks derived from local PoS trigram contexts (Schmid and Atterer, 2004).

In Ananthakrishnan and Narayanan's study, syntax is provided by PoS tags; lexical evidence is represented via syllable tokens {*tea*: `'dx_iy'`; *can*: `'k_aa_n'`; *state*: `'s_t_ey_t'`} transcribed with ARPABET symbols; and acoustic features are encoded as multi-dimensional vectors for each syllable. They find that lexical syllable tokens, augmented with canonical stress labels derived from an open source pronunciation lexicon, are effective for accent detection but not for boundary prediction. Their best phrase break classifier omits this feature stream and achieves 91.61% agreement on the boundary detection task. However, as with Koehn *et al.*, this model incorporates information (in the form of acoustic features) not generally available for TTS systems which take plain text as input.

3.8. Summary

This is a survey chapter and covers key research in automated phrase break prediction as background to this thesis, and how phrase break models are evaluated. Liberman and Church's chink-chunk algorithm, and Taylor and Black's *Festival* model emerge respectively as exemplars for rule-based and statistical approaches to the task in hand. The main finding in this chapter, after detailed analysis of existing feature sets, is the dearth of prosodic features in state-of-the-art phrase break models to complement traditional text-based and syntactic features. This has informed the main hypothesis in this thesis, namely that inclusion of prosodic features may improve the performance of such models, and that real world knowledge of prosody can be represented in a similar way to real world knowledge of syntax: via categorical or descriptive labels.

Chapter 4
Early Experimental Work with Rule-based Models

4.1. Overview

This chapter records early experimental work, and more importantly insights gained which led to hypothesis formulation. Experiments involved two rule-based phrase break models using shallow syntactic features in the form of PoS tags and implementing a shallow parse via NLTK's chunk parser, on the basis that prosodic phrasing is simpler, shallower and flatter than syntactic structure. Both models use the same corpus sample from Section A (Commentary) of the Aix-MARSEC dataset: A08 (annotated by Williams) and A09 (annotated by Knowles). The earlier model is tested on this sample; the later model is developed on short extracts from the sample, plus Williams' boundary annotations in Section C (Lecture: general audience), and then tested on largely unseen text: Knowles' boundary annotations for the remainder of Section C. There is a small amount of deliberate overlap in corpus annotation in Section C to gauge inter-annotator agreement.

4.2. The prepositional phrase model

Intuitive phrasing of Terry Winograd's sentence (§1.6.1; 1.6.2) elicited a couple of options:

The two phrase version:

In the popular mythology **||** the computer is a mathematics machine **||**

The three phrase version:

In the popular mythology **||** the computer **|** is a mathematics machine **||**

It is the author's contention, based on cumulative, native speaker insight into the English language, that the boundary separating the prepositional phrase *in the popular mythology* from the main clause *the computer is a mathematics machine* is more important than the optional boundary between subject and predicate. This is backed up by experimental evidence from the CART statistical model referred to in Chapter 3.3. It was decided therefore to see how far the beginnings and ends of prepositional phrases coincided with boundary annotations by two expert linguists in the aforesaid extracts from Aix-MARSEC.

4.2.1. Research questions

The initial research question, namely *to what extent prosodic phrase boundaries can be located via a major syntactic grouping like prepositional phrases,* was complemented by other questions discussed in the following sub-sections: 4.2.1.1 to 4.2.1.3.

4.2.1.1. To what extent does shallow parsing reflect prosodic phrasing?

The version of NLTK used for this model (nltk_lite version 0.6.5) included a regular expression chunk parser, with accompanying tutorial notes explaining how chunk parsing creates flat '*…structures of fixed depth (typically depth 2)…*' (Bird *et al.*, 2006) and why it is more robust than full parsing. This description ties in with observations about the relative simplicity of prosodic structure and led to the realization that since this method uses regular expressions over PoS tags to chunk *non-overlapping linguistic groupings* in text, it could be used to identify prosodic phrases. There is also the tradition of shallow parsing used to capture prosodic phrasing in the durable *chinks 'n' chunks* algorithm. It was decided therefore to use nltk_lite's chunk parser to set up a rule which specifies prepositional phrases as the node label for chunks and to run this over extracts from the corpus.

4.2.1.2. Can any underlying principles be discovered governing the distribution of minor and major boundaries?

The Aix-MARSEC corpus differentiates minor and major prosodic phrase boundaries (break indices 3 and 4) in an easily detectable, straightforward manner and facilitates comparison between expert annotators. It was anticipated that analysis of the planned chunk parsing experiment would naturally lead to close scrutiny of corpus annotations so that interesting correspondences between prepositional phrases and boundary type might be observed. The discovery of such linguistic patterns in speech corpora and the subsequent process of encoding that new knowledge as rules in a computational model of prosody is an example of what Huckvale (2002) advocates as the practice and goal of speech science.

4.2.1.3. To what extent do people agree on prosodic phrasing?

This is an open-ended question. However, as part of this experiment, the plan was to compare the author’s intuitive prosodic phrasing of extracts used to that of expert annotators’. To accomplish this, plain text versions of the two complete informal news commentaries were obtained; these cover mid-1980s political issues in the Middle East (A08) and South Africa (A09).

4.2.2. Experimental procedure

Preparatory stages in this experimental work cover some of the natural language processing tasks essential to a Text-to-Speech synthesis system, in particular the task of morphosyntactic analysis: assigning part-of-speech tags to word tokens and imposing a hierarchical structure on sequences of PoS tags. However, this hierarchical structure is not a full syntactic parse but a partial chunk parse which only seeks to identify *one* syntactic grouping: prepositional phrases. The experiment assesses the degree of correspondence between the beginnings and ends of prepositional phrases retrieved via the chunk parse rule and “gold standard” prosodic boundary annotations in the Aix-MARSEC Corpus.

4.2.3. The first step: PoS tagging

The chunk parsing experiment and the comparative study of intuitive prosodic phrasing versus boundary annotations in the corpus have both been run using unpunctuated text i.e. no { . , : ; ? () } as well as plain text versions with just the full stops restored. To obtain selected transcripts, the *TextTier* was extracted from the following Notepad files in Aix-MARSEC, available in TextGrid format ready for use with Praat (Boersma and Weenink, 2009): A0801B to A0805B, annotated by Briony Williams and totalling 619 words, plus A0901G to A0906G, annotated by Gerry Knowles and totalling 789 words. Changes to A08 in preparation for PoS tagging with the Brown corpus tagset were as follows:

- *tee double u ay* (airlines) was changed to TWA;
- hyphens were inserted for *x-ray***,** *x-rayed* and *check-in*;
- enclitics such as *that’s* and *they’ve* were restored and all apostrophes checked and left in place e.g. *Shi’ite* and *hero’s*;
- subject-verb agreement was corrected in the following context: ‘*...hijacking**s** from Ben Gurion...are unknown...*’

There are no changes to report for A09, except to say that all apostrophes were checked and left in place e.g. *nobody else's*.

Plain text versions of A08 and A09 were PoS tagged using a composite tagger similar to the one outlined in the nltk_lite tutorial on categorizing and tagging words (Bird *et al.*, 2009, Chapter 5). This takes the form of a bigram tagger trained on tagged extracts from the Brown corpus as "gold standard" (genres A and B, *Press Reportage* and *Press Editorial* respectively); the bigram tagger backs off to a unigram tagger trained on the same genres, which in turn backs off to a default tagger that tags everything as NN, a singular noun. Sample code listing for this, only slightly modified from the original nltk_lite tutorial notes (*ibid.*), is given below and demonstrates the degree to which this toolkit is customised to NLP tasks. Here, the toolkit provides a tokenize() function, various classes of tagger and an associated train() method to facilitate the process of PoS-tagging any input text.

```
text = sourcefile.readlines()
# the next line stores the input text as a list of word tokens in the
variable: tokens
tokens = list(tokenize.whitespace(text))
my_tagger = tag.Default('nn')
unigram_tagger = tag.Unigram(backoff=my_tagger)
train_sents = list(brown.tagged(['a', 'b']))
unigram_tagger.train(train_sents)
bigram_tagger = tag.Bigram(backoff=unigram_tagger)
# the next line trains the tagger on "gold standard" tagged text from the
Brown Corpus
bigram_tagger.train(train_sents)
# the next line stores a new version of the input text as a list of
('token', 'tag') tuples in the variable: tagged
tagged = list(bigram_tagger.tag(tokens))
```

Listing 4.1: Adaptation of NLTK code for constructing and training a composite tagger

The combined tagger correctly tagged 86.13% of word tokens for Aix-MARSEC A08, and 87.07% of word tokens for A09. The tagged versions of Aix-MARSEC were then hand-corrected and all the tags were capitalised ready for the chunk parser. Roughly half the tagging errors resulted from the default tagger (*e.g.* 'past' tagged as NN in the following phrase 'in the past two years'). Significantly, 16.28% of tagging errors in A08 and 21.57% of tagging errors in A09 were due to the word class of prepositions which could be tagged <IN>, <RP>, <RB>, <CS> (preposition, adverb particle, adverb or subordinating conjunction). This had repercussions for the chunk parse rule which specifies a preposition <IN> as chunk node; and it is often difficult to determine whether there *is* an error or not e.g. ‘on’ in ‘…Pretoria’s hold on the mineral rich territory…’ tagged as <RP>.

4.2.4. Developing the chunk parse rule

The chunk parse rule used in this experiment was developed over several iterations on a complex test sentence of 77 words (Paulin, 2003). I have called this the imported rule. Though still a prototype, this rudimentary, catch-all formula attempts to specify the syntactic constituents of any prepositional phrase via a tag pattern, a regular expression pattern over strings of tags delimited by angled brackets and is evidently transferable from one context to another with very little intervention. The only significant changes between the imported rule and versions A08 and A09 are that:

- coordinating conjunctions <CC> have been removed from the rule because they interfere with boundary prediction (see discussion in Section 5);
- as a stop-gap measure, <PP$> (personal pronoun: possessive) has been replaced by <POSS> (a made-up tag) simply because the chunk parser does not recognize the dollar symbol.

4.2.4.1. Imported rule version

The tag pattern and description string for this rule instruct the parser to begin the chunk with a word token tagged as a preposition, and to include in that chunk *any* combination in *any* order of tokens tagged as follows: another preposition; determiner/pronoun (singular); determiner/pronoun (singular or plural); article; personal pronoun (object); nominal pronoun; determiner/personal pronoun (possessive); adjective; coordinating conjunction; noun (singular); noun (plural).

```
parse.ChunkRule('<IN><IN|DT|DTI|AT|PPO|PN|PP$|JJ|CC|NN|NNS>+',
"Chunk prepositions <IN> with sequences of prepositions <IN>
determiners <DT|DTI>  articles <AT> object or nominal pronouns
<PPO|PN> possessive determiners <PP$> adjectives <JJ> coordinating
conjunctions <CC> and common nouns <NN|NNS> using the Brown
tagset")
```

Listing 4.2: Prepositional chunk node and description string for initial ruleset

4.2.4.2. A08 rule version

This rule removes <CC> (coordinating conjunctions), replaces <PP$> with <POSS>, and adds the following constituents: determiner/pronoun or post determiner; cardinal number; superlative adjective; proper noun.

```
parse.ChunkRule('<IN><IN|DT|DTI|AT|AP|CD|PPO|PN|POSS|JJ|JJT|NP|NN|NN
S>+', "Chunk prepositions <IN> with sequences of prepositions <IN>
determiners <DT|DTI>  articles <AT> determiner/pronoun or post
determiner <AP> cardinals <CD> object or nominal pronouns <PPO|PN>
possessive determiners <POSS> adjectives and superlatives <JJ|JJT>
proper nouns <NP> and common nouns <NN|NNS> using the Brown tagset")
```

Listing 4.3: Prepositional chunk node and description string for Chunk Parse 1 (§4.2.6)

4.2.4.3. A09 rule version

This rule incorporates the following additions: ordinal numbers and semantically superlative adjectives.

```
parse.ChunkRule('<IN><IN|DT|DTI|AT|AP|CD|OD|PPO|PN|POSS|JJ|JJT|JJS|N
P|NN|NNS>+', "Chunk prepositions <IN> with sequences of prepositions
<IN> determiners <DT|DTI>  articles <AT> determiner/pronoun or post
determiner <AP> cardinals <CD> ordinals <OD> object or nominal
pronouns <PPO|PN> possessive determiners <POSS> adjectives <JJ>
superlatives <JJT> semantically superlatives adjectives <JJS> proper
nouns <NP> and common nouns <NN|NNS> using the Brown tagset")
```

Listing 4.4: Prepositional chunk node and description string for Chunk Parse 2 (§4.2.6)

4.2.5. Intuitive prosodic phrasing

A further aspect of this experimental work, and a means of familiarisation with the corpus, was to compare the author’s intuitive prosodic phrasing to that of expert annotators’ and to mark out longer prosodic phrases in response to Liberman and Church’s own criticism of the chink chunk rule in their original paper. They consider the prosodic phrases or ‘*function word groups*’ captured by the rule to be too small to accommodate sufficient variation in prosody and are interested in discovering how these smaller units ‘…*combine hierarchically to form sentence-sized units*…’ The procedure followed in the current study was to assign major and minor boundaries with the same pipe symbol notation as the corpus, using unpunctuated text versions of A08 and A09 (*i.e.* no commas or full stops *etc*) and without reference to the original recordings. Intuitive boundary locations and types were then compared to corpus annotations. An example of these intuitive predictions is given below and set alongside corpus annotations in a short extract from A08 where the phrasing is quite dense – more so in the author's version than the original. The intuitive phrasing version also arranges the text so that what are considered to be the most important boundaries, those giving rise to longer prosodic phrases, appear at the end of the line:

Intuitive phrasing:
Given the state of lawlessness that exists in Lebanon ||
the uninformed outsider | might reasonably expect | security | at Beirut airport |
to be amongst the tightest in the world ||
but the opposite is true ||

Example 4.1: Intuitive phrasing variant for sample sentence from the corpus

Corpus annotations:
Given the state of lawlessness that exists in Lebanon || the uninformed outsider might reasonably expect security | at Beirut airport || to be amongst the tightest in the world || but the opposite is true ||

Example 4.2: Corpus annotation of minor and major phrase boundaries by Williams

4.2.6. Results

The chunk parser's rule-based identification of prosodic phrases via retrieval of prepositional phrases, plus the author's intuitive predictions were compared to "gold standard" boundary annotations of extracts A08 and A09 in the Aix-MARSEC corpus by two expert linguists. An overview of how many boundaries of both types (major and minor) were correctly located by rule and by human judgement is presented in this section in Tables 4.1 and 4.2, while the discussion of error types – deletions (missed boundaries) and false insertions – is continued in Chapter 5.2.

	BW "Gold Standard"	Chunk Parse 1	Chunk Parse 2	Intuitive Phrasing
Total number of boundaries minor +	120	not run	110	93

major				
Total number of boundary positions correct			56	85
Total number of major boundaries	67			60
Total number of major boundaries correctly located			33	45
Total number of minor boundaries	53			33
Total number of minor boundaries correctly located			23	12
Total number of full stops	33			33
Total number of full stops correctly located				32

Table 4.1: Raw counts for boundaries retrieved by rule and human judgement (A08)

	GK “Gold Standard”	Chunk Parse 1	Chunk Parse 2	Intuitive Phrasing
Total number of boundaries minor + major	200	131	135	156
Total number of boundary positions correct		81	87	139
Total number of major boundaries	31			52
Total number of major boundaries correctly located		9	18	31
Total number of minor boundaries	169			104
Total number of minor boundaries correctly located		72	69	83
Total number of full stops	24			24
Total number of full stops correctly located		7	15	23

Table 4.2: Raw counts for boundaries retrieved by rule and human judgement (A09)

4.2.7. Initial reflections

In evaluating the effectiveness of the chunk parse rule and the intuitive phrasing approach, 3 different measures have been used: total number of boundary positions correctly located; number of major and minor boundary types correctly located; and number of full stops correctly located. The first measure does not distinguish between major and minor boundaries; so as long as boundary site was correctly identified, an exact match between position and boundary type was not looked for. Chunk parse 1 took as input text without full stops or commas *et cetera* (as did the author when making intuitive predictions) but this did not locate boundaries where constituents included in the rule spanned the boundary as in:

‘…some form **{**of local government || at a news conference**}**…the party leaders…’

Example 4.3: Two consecutive prepositional phrases spanning a sentence boundary

This approach was therefore abandoned, with an overall success rate of 40.50% boundary positions correctly located in A09. For Chunk parse 2, full stops *only* were restored and this gave marginally better performance: 43.50% boundary positions correct for A09 and 46.66% correct for A08. Obviously, detection could be improved with fuller punctuation but as already pointed out, punctuation is partly a matter of style and the idea behind this experiment was to create a catch-all rule, independent of text domain.

Syntactic contexts in which the chunk parse rule does seem to approach natural phrasing include consecutive prepositional phrases, for example:

‘…**{**near the top of the political agenda of the major Western powers**}**…’

Example 4.4: Rule captures syntactic dependencies between prepositional phrases

One could argue for a boundary after the word ‘agenda’; equally, one could get by quite comfortably without it. The chink chunk rule would create a surplus of boundaries here – 3 in all. This example does raise one issue, however, about the status of the preposition *‘of’* which seems to have a weaker semantic identity than other prepositions and which is reliant on neighboring nouns. Here, the word *‘of’* marks degrees of proximity to a desired target: the TOP of a particular agenda. Its link-up role can be illustrated by a further example where a boundary is invoked at the point where *‘of’* re-establishes contact between target and tributary nouns in the pattern ‘…*a picture of*..:’

‘…an x-ray picture **|** on two TV screens **||** of the contents of hand baggage…’

Example 4.5: Prosodic-syntactic boundary agreement in Williams’ annotations

Corpus annotations indicate the boundary after ‘*screens*’ is stronger than the boundary after ‘*picture*’.

4.2.7.1. Reflections on intuitive prosodic phrasing

Perhaps the most interesting result of this three-way comparison of predicted and perceived prosodic phrasing is *within-sentence* allocation of major boundaries by the author and by Knowles and Williams. Raw counts from Tables 4.1 and 4.2 have been reworked in Table 4.3.

	% major boundaries not accounted for by full stops			
	GK	**CB**	**BW**	**CB**
A09	22.58%	53.85%		
A08			50.75%	45%

Table 4.3: Author preferences seem closer to Williams' allocation of major phrase boundaries

The further point of interest is the performance of this rather crude chunk parse rule relative to human judgement. The former gets between 43 and 47 per cent of boundaries correct for A09 and A08 respectively, while the latter scores between 69 and 71 per cent. The rule-based method actually performs *better* than the author when discovering minor phrase boundaries in A08.

4.3. The stoppers and starters model

The prepositional rule model tries to identify likely constituents of prepositional phrases and its focus is therefore *inside* prosodic units. A second model, nicknamed the *stoppers and starters* model, takes a different approach. As with CFP rules, this model differentiates between PoS that generally terminate prosodic units (*i.e.* the stoppers or chunks) and PoS that generally initiate new ones (*i.e.* the starters or chinks); but it also exploits the fact that some PoS occur in either position, and prompts the following question:

Instead of a binary division into content and function words, could we infer four groupings: stoppers, starters, *both-ers* (*i.e.* dual-functioning PoS as terminators and initiators of prosodic units) and *neith-ers* (*i.e.* medial components of prosodic units)?

4.3.1. The two-stage chunker

It is possible to apply more than one chunk pattern using NLTK's regular expression based chunk parser. In a two-stage (or multiple-stage) chunker (Bird *et al.*, 2009, Chapter 7) rule ordering is important since the subordinate rule will only create new chunks in material that is already partially chunked if there is no overlap. The prototype stoppers and starters model is a two-stage chunker, developed iteratively, which attempts to:

- focus on the boundary itself, rather than the contents of prosodic chunks;
- differentiate between boundaries preceded by nouns, adjectives, and certain pronouns as terminators (the dominant rule) versus boundaries preceded by other parts of speech (the subordinate rule).

By way of illustration, an early version of this rule explores LOB categories which mostly end up in the dominant rule; earlier code (from 2006) has been updated here for compatibility with a more recent version of NLTK (0.9.8). The code in Listing 4.5 takes liberties with the ChunkRule() method and uses it to isolate boundary positions, formulating the chunk node for each parse not as a major syntactic grouping, such as noun phrases (NP), but as an instruction to insert a boundary. Example outputs are compared to target prosody from the corpus.

Target prosody from A08

...it’s frequently been used | by arab hijackers | as the starting point | for their operations | because it’s such an easy touch | for anyone wanting to smuggle weapons onto an aircraft...

```
import nltk, re
from nltk.chunk.regexp import *
text1 = [("it's", 'PPS+HVZ'), ('frequently', 'RB'), ('been', 'BEN'),
('used', 'VBN'), ('by', 'IN'), ('Arab', 'JNP'), ('hijackers', 'NNS'),
('as', 'IN'), ('the', 'ATI'), ('starting', 'JJ'), ('point', 'NN'), ('for',
'IN'), ('their', 'POSS'), ('operations', 'NNS'), ('because', 'CS'),
("it's", 'PPS+BEZ'), ('such', 'ABL'), ('an', 'AT'), ('easy', 'JJ'),
('touch', 'NN'), ('for', 'IN'), ('anyone', 'PN'), ('wanting', 'VBG'),
('to', 'TO'), ('smuggle', 'VB'), ('weapons','NNS'), ('onto', 'IN'), ('an',
'AT'), ('aircraft', 'NN'), ('.', '.')]
domRule = ChunkRule('<NN|NNS|JJT|NP|JNP|RP|RB><IN|OF|CS|CC>+',
'Oppose  sequences of nouns and other types that behave as stoppers
with starters like prepositions and conjunctions')
subRule = ChunkRule('<NN|NNS|NP|JNP|JJT|PN><VBG|PPSS>', 'Oppose
sequences of nouns and other types that behave as stoppers with
collapsed relative clauses introduced by present participles or
reflexive pronouns')
chunker = RegexpChunkParser([domRule, subRule], chunk_node =
'INSERT_BOUNDARY')
output = chunker.parse(text1)
print output

(S
  it's/PPS+HVZ
  frequently/RB
  been/BEN
  used/VBN
  by/IN
  Arab/JNP
  (INSERT_BOUNDARY hijackers/NNS as/IN)
  the/ATI
  starting/JJ
  (INSERT_BOUNDARY point/NN for/IN)
  their/POSS
  (INSERT_BOUNDARY operations/NNS because/CS)
  it's/PPS+BEZ
  such/ABL
  an/AT
  easy/JJ
  (INSERT_BOUNDARY touch/NN for/IN)
  (INSERT_BOUNDARY anyone/PN wanting/VBG)
  to/TO
  smuggle/VB
  (INSERT_BOUNDARY weapons/NNS onto/IN)
  an/AT
  aircraft/NN
  ./.)
```

Listing 4.5: Boundary annotations **in bold** from two-stage chunker match *gold standard* corpus annotations

Although the output does not exactly match corpus phrasing (the chunker gets 4/5 boundaries correct but also inserts 2 others), and although the rules do not incorporate automatic boundary insertion after a full stop, there is nothing wrong with the output. We will return to this in the next chapter.

4.3.2. Prototype grammar

The chunk parser outlined in this section implements (i) a dominant rule which retrieves boundaries between nominal, adjectival and some pronominal categories versus *other parts of speech*; and (ii) a subordinate rule which then intuitively juxtaposes likely stoppers with likely starters *in* those remaining other parts of speech. The tag pattern is given largely by sequences of LOB tags (*i.e.* the chunker uses a more discrete tagset). For the dominant rule, the description string emphasises *pre*-boundary items whereas in the subordinate rule, the description string emphasises *post*-boundary items.

Using NLTK's ChunkRule() method, the dominant rule for the current prototype in Test 3 (§4.3.4) effectively inserts a boundary after nouns, nominal and reflexive pronouns – *and* superlative adjectives because they can behave like nouns.

In addition, its subordinate rule opposes likely stoppers with the following clause markers: existential *there*, coordinating and subordinating conjunctions, prepositions, the word *not*, WH-pronouns, the word *to* denoting an infinitive of purpose, WH-determiners, and a made up tag <ATNEG> for negative forms of the article as in *no*.

```
domRule =
ChunkRule('<N.*|JJT|JNP|PN|PPL|PPLS><ATNEG|WPOSS|EX|CC|CS|IN|XNOT|TO
|MD|VBN|VBG|PPSS|POSS|VB|VB.*|BE|BE.*|HV|HV.*|WPO|WPS|WDT|WRB|RB|PPS
BEZ|EXBEZ|DTBEZ>',
'Effectively insert a boundary after the same list of noun,
adjective and pronoun types as in Test 2, but using RE operators to
simplify presentation somewhat')

subRule =
ChunkRule('<WRB|RP|PPS|PPSS|PPL|RB|VBN|VBZ|VB|VBD|BEM|PPLS|CS><EX|CC
|CS|IN|XNOT|WP|WPS|WPO|TO|WDT|ATNEG>',
'Oppose likely stoppers, adding subordinating conjunctions <CS> to
this list - with clause markers including 2 new additions: WH-
determiners and a made up tag <ATNEG> for negative forms of the
article as in NO')
```

Listing 4.6: Optionality in chunk node for dominant rule and identification of further chunk nodes in relation to major clause markers in subordinate rule

4.3.3. Example outputs

The stoppers and starters rule recognises potential boundary sites via PoS tag oppositions (in effect unweighted bigrams) observed from empirical data in Aix-MARSEC. The outputs themselves are similar to those of other CFP algorithms in that they capture low level phrasal units; but the rule is also able to match corpus phrasing which discriminates *between* words that are sometimes classed as function words (see **bold** items in Example 4.6 below), one objective of model design being to explore the conventional mapping of function words to chinks.

… cast/VBN their/POSS spell/NN | not/XNOT **only/RB | on/IN** our/POSS eminent/JJ professional/JJ colleague/NN Dr/NPT FitzGerald/NP | but/CC **also/RB | on/IN** Mr/'NPT Howell/NP | who/WP **himself/PPL | has/HVZ** a/AT First/OD Class/NNP degree/NN | in/IN Economics/NNP …

Example 4.6: Pronouns are often classed as function words, so too some forms of adverbs (*e.g.* particles), and verbs which function as auxiliaries (*cf.* the *chinks ‘n’ chunks* algorithm)

Raw preditions in Example 4.7 show the rule working quite well on a sentence from the Reith Lecture transcript; the annotator here is Briony Williams. True positives (boundaries correct) are marked ☑ and false positives (false insertions) marked ☒. Commas were deliberately stripped from input text and comma sites retrieved by the rule at major chunking boundaries are therefore given in bold.

```
('one', 'CD')
(INSERT_BOUNDARY: ('aspect', 'NN') ('of', 'IN')) ☒
('this', 'DT')
(INSERT_BOUNDARY: ('centralism', 'NN') ('is', 'BEZ')) ☑
('the', 'ATI')
(INSERT_BOUNDARY: ('idea', 'NN') ('which', 'WP')) ☑
('has', 'HVZ')
('been', 'BEN')
(INSERT_BOUNDARY: ('embraced', 'VBN') ('by', 'IN')) ☒
('successive', 'JJ')
('British', 'JNP')
(INSERT_BOUNDARY: ('governments', 'NNS')☒
('of', 'IN')) □
('both', 'ABX')
(INSERT_BOUNDARY: ('parties', 'NNS') ('that', 'CS')) ☑
('a', 'AT')
(INSERT_BOUNDARY: ('choice', 'NN') ('has', 'HVZ')) ☑
('to', 'TO')
('be', 'BE')
(INSERT_BOUNDARY: ('made', 'VBN') ☑
('at', 'IN')) □
('Cabinet', 'NP')
(INSERT_BOUNDARY: ('level', 'NN') ('of', 'IN')) ☑
('one', 'CD1')
('particular', 'JJ')
('reactor', 'NN')
(INSERT_BOUNDARY: ('system', 'NN') ('for', 'IN')) ☑
('future', 'NN')
('nuclear', 'JJ')
('power', 'NN')
(INSERT_BOUNDARY: ('stations', 'NNS') ('in', 'IN')) ☑
('Britain', 'NP') ('.', '.')
```

Example 4.7: Phrase break predictions from this rudimentary two-stage chunker retrieve sites of commas at major clause boundaries

4.3.4. Concluding Comments

NLTK recommends several rounds of rule development and testing in order to create a good chunker. The data in Examples 4.6 and 4.7 was obtained by running the simple chunking algorithm on part of the Reith Lecture transcript annotated by Briony Williams (1463 tokens); manually examining outputs and refining the rule;

and running the revised rule on the same section and finally on a previously unexamined section - i.e. the remainder of the Reith Lecture transcript annotated by Gerry Knowles (2445 tokens). Scores were recorded as shown in Table 4.4.

	Annotator	**Precision %**	**Recall %**	**F-score %**
Test 1	BW	65.19	59.03	61.96
Test 2	BW	66.76	71.35	68.98
Test 3	GK	70.85	61.04	65.58

Table 4.4: Sample P, R and F-scores from development tests on chunk parse phrase break rule based on unweighted bigrams

4.4. Summary

This chapter gives an account of early experimental work which constitutes part of the learning process in this thesis, and contributes to hypothesis formulation. The prepositional model (§4.2) and the stoppers and starters model (§4.3) are both incomplete projects, but have been included for the following reasons:

1. They evidence a shift in emphasis in this thesis from attempting to define the *contents* of tone groups or prosodic phrases to concentrating on the boundary itself, that is the attributes – *including prosodic attributes* – of lexical items adjacent to the boundary.
2. Their movement value in re-thinking the binary divide between content and function words for phrase break prediction which, for example, informs syntactic and prosodic feature extraction for machine learning experiments in Chapter 10 (§10.4 and 10.5).
3. Consideration of outputs from both models have prompted closer inspection of corpus annotation in SEC and Aix-MARSEC, and given rise to motivational insights into prosodic variance which are discussed in the next chapter.

Chapter 5
The Variability of Prosodic Phrasing

5.1. Overview

In this chapter, the limitations of only using syntactic and text-based cues for phrase break prediction are further discussed, with evidence from the corpus, plus outputs from the prepositional phrase model (Chapter 4.2) in the form of false positives (extra insertions) and false negatives (missed boundaries) used as illustration. The discussion then turns to the limitations of evaluating any phrase break model against a "gold standard" which itself only represents one phrasing variant for an utterance or text. Again, there is detailed consideration of early experimental results in the form of predictions from the stoppers and starters model (Chapter 4.3), and evidence from the corpus, in support of the argument.

5.2. Why syntax is not enough: evidence from the corpus

Shallow or chunk parsing is a common methodology associated with phrase break prediction; there is consensus that prosodic phrasing is somehow simpler and flatter than syntactic structure. Hence *chink-chunk* or CFP-type algorithms are still used to identify low-level phrasal units in TTS (Knill, 2009). Noun phrase (NP) chunks are also represented in terms of IOB tags (Ramshaw and Marcus, 1995; Bird *et al.*, 2009, Chapter 7) where word tokens are classified as constituents (inside) or non-constituents (outside) of NPs or as initiating (beginning) NPs. Hence, "beginners" correspond to chinks: closed-class or function words immediately preceded by an open-class or content word – the signal for boundary insertion (Liberman and Church, 1992).

5.2.1. Inside or outside the chunk?

The prepositional phrase model, which attempts to define likely constituents of *prepositional* phrases using a chunk parser from the Natural Language ToolKit, demonstrates the shortcomings of such catch-all rules. The examples in Table 5.1 from Section A08 (1) and A09 (2-4) of our development set in Aix-MARSEC show prepositions (in **bold**) beginning (*e.g.* 2) or not beginning (*e.g.* 1) a prosodic phrase, as the speaker decides. Moreover, some forms elude placement inside or outside the prepositional phrase chunk.

1	**on** aeroplanes **|** *flying* **around** the Middle East and
2	**| on** top **of** a hill **|** *overlooking* Windhoek **|**
3	which French authorities **|** had made in their *handling* **of**
4	fly back **to** South Africa **|** *leaving* those **|** internal leaders **|**

Table 5.1: Prepositions and particles, plus gerunds and participles are difficult to categorise for prosodic-syntactic boundary placement.

Resolving the problem in (3) would be a straightforward case of re-tagging the word *handling* as a gerund or verbal noun and identifying this new tag as a likely constituent of prepositional phrases. Examples (2) and (4) could not be resolved so easily: we can imagine a legitimate amalgamation of the prosodic chunks in (2) and might wish to retain the option of including participles within prepositional phrase sequences; we would not want this option in (4) however, where the participle initiates a new syntactic chunk and has nothing to do with the prepositional phrase. Finally, what do we make of the chopped-up NP *those internal leaders* in (4)?

5.2.2. Category blends

Manning and Schutze (1999, p.12) discuss ambiguity caused by non-categorical behaviour of parts of speech: individual words can be PoS-tagged differently in different syntactic contexts and, though allocated a particular PoS tag in a particular context, may retain and exhibit simultaneous behaviours *e.g.* "-ing" forms blurring the distinction between nouns and verbs. Another blurred category where word forms lean towards "left" (outside) or "right" (inside) behaviours relative to prosodic boundaries is particle <RP> versus preposition <IN> respectively. Such "tagging" is fluid in spontaneous speech (§5.3.5 and 5.3.6).

The prepositional rule inserts a boundary before true prepositions, PoS-tagged <IN>. There are six items tagged as true prepositions in the snippets in Table 5.1 and only one particle: "*back*" in (4). However, there does not seem to be much difference between the preposition-particles "*flying around*" in (1) and "*fly back*" in (4); and the absence of boundaries in speakers' chunking gives particles the benefit of the doubt here.

5.2.3. Rhythmic clout

As yet, we do not have a definitive set of content-function word groups mapped to parts-of-speech. The lexicon discussed in Chapter 6 uses the same default mappings of CF to Penn Treebank tags as Busser *et al.* (2001) and Bell (2005). Nevertheless, we are likely to be in accord about the CF category labels allocated to the sentence fragment in row 1 of Table 5.2.

1	F	F	C	F	F
2	before	the	hijacking	of	the
3	ANA+NRU	ANA	NRU	ANA	

Table 5.2: Binary classifications for syntax and rhythm

Row 3 represents rhythmic annotations from the Jassem Tier (Bouzon and Hirst, 2004) in the Aix-MARSEC dataset. The label NRU (*narrow rhythm unit*) denotes either a stressed syllable in a monosyllabic word or a stressed syllable followed by a number of unstressed syllables in a bi-syllabic or polysyllabic word, while the label ANA (*anacrusis*) denotes an unstressed word-initial syllable or a sequence of unstressed syllables unattached to any NRU. Syntactically, the word *before* behaves as a function word in this example but rhythmically it shares attributes with content words, carrying a beat (primary stress) on a long vowel.

A similar situation arises if we view the whole of this opening sentence in A08.

A few days before the hijacking | of the TWA aircraft | soon after it took off from Athens airport || I was catching a similar TWA flight | from the same airport. ||

Here we have two instances of the preposition “*from*” – another grammatical or function word – which have different phonetic and rhythmic properties. We can verify this by inspecting instances from the TextGrid file for section A0801 in Aix-MARSEC, as presented in Table 5.3.

Tonic Stress Marks Tier	Jassem Tier

`5.0099999999999998`
`"from"`
	`5.0099999999999998`
	`"ANA"`

5.0099999999999998	5.0099999999999998
9.1639999999999997 "~from" 9.1639999999999997	9.1639999999999997 "NRU" 9.1639999999999997

Table 5.3: Even grammatical words exhibit prosodic variance

Vowel reduction in the first occurrence of /fr@m/ makes it an anacrusis. Conversely, the second instance of /frQm/ is a narrow rhythm unit and even carries a pitch accent.

5.2.4. Taking stock

In summary, our example sentence exhibits all sorts of recalcitrant prosodic-syntactic behavior. A syntax-based rule which inserts a boundary before true prepositions or between content and function words, or between major syntactic groupings (NP/AVP: *A few days* versus PP: *before the hijacking*) is insensitive to speaker evidence here, where the adverbial qualifier is being treated prosodically as part of the prepositional phrase chunk since its role is to enhance the specificity of that phrase.

5.3. Why "gold standard" evaluation of prosody is problematic: evidence from the corpus

Two publications discussed in this thesis raise questions about the practice of evaluating a prosodic phrase break model against a gold standard; in both cases the iconic prosodic annotations in versions of the Spoken English Corpus. Taylor and Black (1998) state that performance figures obtained in such experiments should be '…treated with caution…' because prosody itself is subjective: different speakers pause in different places; one speaker will vary their use of pauses; expert annotators differ in their perceptions. Similar comments about variability in human performance appear in Hirschberg (2002). Taylor and Black also point out that junctures differ in type: those junctures which coincide with weaker *syntactic* boundaries are more likely to be potential *prosodic* boundary sites (see also Abney, 1992; Abney, 1995). Knowles points out that, having established a dual-level boundary annotation set for SEC, transcribers duly '…interpret [their] observations as realisations of members of the set of categories…' (Knowles, 1996b, p.88), but then encounter '…several different patterns…subsumed under a category like tone group boundary…' (*ibid*, p.94). In addition, Pickering *et al* (1996, p.67) note that Knowles perceives shorter tone units than his counterpart, Williams; this is further evidenced in this thesis in Section 5.3.1. Moreover, despite claims (*ibid*, p. 67) and evidence of inter-annotator agreement on boundary type in overlapping corpus annotations, these overlapping sections are few and far between, and we have found evidence that: (i) Williams makes bolder use than Knowles of the major boundary marker *within* sentences when she is sole transcriber (*cf*. Examples 4.2 and 4.5 in the previous chapter); and that: (ii) different boundary types have been assigned to similar syntactic contexts (*cf*. the discussion in Section 5.3.3 of both minor and major *prosodic* boundaries mapped to major clauses). Atterer and Klein (2002) encapsulate all these reservations and dichotomies: '…the very notion of evaluating a phrase-break model against a gold standard is problematic as long as the gold standard only represents one out of the space of all acceptable phrasings…'

5.3.1. Inter-annotator agreement

The 'spaciousness' of acceptable prosody can be demonstrated straightaway by the gold standard itself in Examples 5.1 and 5.2, a sample from Section C in Aix-MARSEC. The extract comes from a Reith Lecture and is illustrative because, while there is only one speaker, there are two alternative phrasings: this is one of the overlapping sections of prosodic annotation from Briony Williams and Gerry Knowles (approximately 9% of the corpus).

The main difference between Knowles' and Williams' boundary annotations here seems to be one of perception. In the section marked in **bold**, Gerry Knowles 'hears' a more emphatic speaker than Briony Williams and inserts more pauses overall (35 instead of 29). Both annotators insert a boundary at every punctuation mark in the original raw text transcript - another acceptable phrasing, perhaps?

for some people | this statement of orthodox economic doctrine | may appear | too unqualified || since it fails to mention explicitly | security of supply || often | though not always | the case for self sufficiency is argued | with reference to a country's need to ensure security | by minimising dependence | on foreign sources || the outside world is seen | at best | as unreliable | and subject to instability | at worst | as actively hostile || **from this fortress mentality | standpoint | autarchy | appears | to be common prudence ||** two sets of measures | then suggest themselves | one is to build up | domestic production of essentials | so as to reduce imports | to a minimum | the other | is to restrict exports | so as to ensure | that domestic supplies | are available | for domestic use ||

Example 5.1: This is a sample of Gerry Knowles' phrase break annotations for a BBC recording of a Reith Lecture from the 1980s

for some people | this statement of orthodox economic doctrine | may appear too unqualified || since it fails to mention explicitly | security of supply || often | though not always | the case for self sufficiency is argued | with reference to a country's need to ensure security | by minimising dependence on foreign sources || the outside world is seen at | best | as unreliable | and subject to instability | at worst | as actively hostile || **from this fortress mentality standpoint | autarchy appears to be common prudence ||** two sets of measures | then suggest themselves | one | is to build up domestic production of essentials | so as to reduce imports | to a minimum || the other | is to restrict exports | so as to ensure | that domestic supplies | are available for domestic use ||

Example 5.2: Briony Williams' phrase break annotations for the same sample in C

Example 5.3 shows both annotators largely in agreement on phrasing and on emphatic, bi-tonal accents (rise-falls) in a snapshot sentence from Examples 5.1 and 5.2. The only area of dispute is whether or not to include a boundary after the word '…dependence…'

,often | though not **`/always |** the case for self sufficiency is **`/argued |** with reference to a country's need to ensure **se`/curity |** by minimising dependence | on foreign sources

Example 5.3: This is the corpus version, showing all prosodic phrase breaks noted by Knowles and Williams and the pitch accent annotations on words preceding boundaries where the experts are in agreement.

However, what if a new speaker took this same text and chunked it differently with the explicit intention of prioritising certain syntactic structures or constituents? What about the new phrasing in Example 5.4, for example, which differs from the original by deliberately highlighting intentions, movements, actions present in verb forms?

often | though not always | the case for self sufficiency | is **argued |** with reference to a country's need | *to ensure security* | by **minimising |** dependence on foreign sources

Example 5.4: This alternative phrasing is largely achieved within the performance structure of the original (§4.3.2)

5.3.2. The space of acceptable phrasings

The emphatic combination of (high) chunking accent and boundary in the matching annotations in **bold** in Example 5.3 is typical of English. An emphasis-boundary pattern has now been engineered in Example 5.4 for the participle **'…minimising |…'** (which gets a high level pitch accent from both annotators) and could enhance the infinitive construction '…*to ensure security*…' if a boundary were to be placed before the noun.

new instance:
[NP a country's need] | [VP to ensure | security]

new instance:
[VP to ensure **|** security] [PP by minimising dependence]

new instance:
[PP by minimising | dependence] [PP on foreign sources]

Example 5.5: Prosodic boundaries are shown in relation to the large syntactic units {NP, VB, PP, ADJP, ADVP} featured in Koehn et al, (2000).

The difference between these new instances and the original template in Example 5.3 is that most of them occur *within* and not *between* discrete syntactic groupings – lower down the tree as it were. This is illustrated in Example 5.5. It is also worth noting that the only disruption these ‘false insertions’ make to the original phrasing surrounds the noun ‘…dependence…’ where Knowles and Williams are not in agreement anyway (Examples 5.1 and 5.2). The new instances in Example 5.5 are not disfluencies (speaker hesitations); in fact, they evidence a coherent strategy on the part of the speaker to emphasise ‘doing’. Furthermore, even though they would be classed as *false insertions* when compared to the corpus gold standard, they are definitely not wrong.

5.3.3. Different boundary types

Of course, that is not the end of the story. A new complication now arises in that these different types of boundaries - the *chunkers* higher up the syntax tree and the *highlighters* lower down the tree - are not differentiated in the corpus. It would be nice if they were analogous to major and minor boundary classifications and the symbols: < || > and < | >. This is not the case, however. Examples 5.6 and 5.7 show the same annotator (in this case Briony Williams) using different phrase break annotations to flag up major clause boundaries in a news bulletin and a lecture. The association of double pipes (ToBI’s break index 4) with major syntactic groupings, plus the use of pitch accent annotations *without boundary reinforcement* for highlighting in the first extract, seems much clearer.

there are ~two \,scanning machines || which give an `/X ray picture | on two tele`/vision *screens || of the _contents of `hand *baggage || when `/I've been through *Athens airport || and `that's about *two dozen `times in the past *two `/years || there's `/never been more than ~one se\,curity man on *duty || and ~he's

\frequently reading a `newspaper || or ~chatting with _other `airport *staff ||

Example 5.6: This is a sample of Briony Williams' annotations of informal news commentary from a BBC radio broadcast from the 1980s. It shows correspondence between major intonation unit boundaries and major clause boundaries.

the `/history | of ~British nuclear ,power programmes | ~over the past thirty ,years | >pro_vides a de~pressing e\xample | of ~unreflecting _centralism in `action || `stoutly rein`forced | _I may /add | by `/other forms | of _DIY`E || `one aspect of this ,centralism | is the **i`/dea | which** has been em~braced by su*ccessive British _governments of `/both **parties | that** a ,choice | `has to be made | at `/Cabinet level | of ~one par,ticular re`/actor system | for _future nuclear `power stations | in \Britain ||

Example 5.7: Another sample annotation from Briony Williams shows minor intonation unit boundaries being used to demarcate major clause boundaries.

5.3.4. Chunking versus highlighting

The examples so far have demonstrated how denser prosodic phrasing (highlighting) can be inserted into the existing chunk structure of a sentence. The rest of Section 5.3 covers instances where prosody redistributes prominence, first by ignoring, and second by shifting chunk boundaries.

True positives retrieved by the stoppers and starters model are given in Example 5.8 and evidence a reliable rule-of-thumb when a major clause boundary and comma-site is retrieved before a subordinating conjunction.

...the idea | which has been embraced by successive British governments of both parties | that a choice | has to be made...

Example 5.8: Phrase break predictions from the rudimentary syntax-driven rule retrieve sites of commas at major clause boundaries.

This is generally a PoS context where prosody, performance structure and syntax (Abney, 1992) are in agreement: a prosodic boundary generally occurs with a major clause boundary. Nevertheless, the mismatch between prediction and empirical evidence in Example 5.9 below shows the speaker making a different chunking choice for this PoS context - glossing over a major syntactic boundary and favouring the highlighting over the chunking function of prosody by placing adjectives **'…important…self-sufficient…'** in phrase-final position. Consequently, predictions-by-rule quickly get out of sync with empirical phrasing (though not out of sync with naturalness) because they each start to take a different processing route through the sentence. As a final twist, however, predicted phrasing manages to regain contact with the original after coverage of the *theme* (everything before the copula) is complete (see **bold** items in Example 5.9).

Corpus phrasing:

'…The idea that it's **important |** for developing countries to become **self-sufficient |** in food | **is** widely | and uncritically accepted | not just in Brussels; | but from the orthodox economic standpoint | it's without foundation…'

Predicted phrasing:

'…The **idea |** that it's important for developing **countries |** to become self-sufficient in food | **is** widely | and uncritically accepted | not just in Brussels | but from the orthodox economic standpoint | it's without foundation…'

Example 5.9: Predicted phrasing matches the corpus once the theme (everything before the copula '…is…') is established.

5.3.5. Prepositions versus verb particles

The prototype rule inserts a boundary before true prepositions, PoS-tagged <IN>. This accounts for false inserts - but legitimate, if somewhat emphatic ('Tony Blair style') prosodic phrasing - in the following sentence fragment in Example 5.10.

Corpus phrasing:

'…the idea | which has been embraced by successive British governments of both parties | that a choice | has to be made…’

Predicted phrasing:

‘…the idea | which has been ***embraced | by*** successive British governments | of both parties | that a choice | has to be made…’

Example 5.10: Predicted phrasing abides by the gold standard POS tagged version of this sentence which classifies the function word ‘…by…’ as a preposition.

It will be noted from Example 4.8 that there are four empirically verified (true positive) phrase boundaries before prepositions in the section as a whole. Moreover, since the PoS-tagged version of this text is itself a gold standard, and since this version classifies ‘…embraced by…’ as <VBN><IN> (a past participle followed by a preposition), we have a situation where two equally valid gold standards - tagged text versus prosodic annotation - are in conflict. This arises because the same speaker in this particular instance has realised ‘…embraced by…’ as one unit and, *via prosody*, has in effect tagged the preposition as a verb particle: <VBN><RP>. This rules out an intervening *chunking* prosodic phrase boundary and significant *chunking* accent on ‘…embraced…’ Corpus annotation on the verb testifies to this: **em~braced** is a level accent.

5.3.6. A conflict of standards?

Abney (1991) raises the thorny issue of prepositional phrase attachment, ‘…the most explosive source of ambiguity in parsing…’ The PoS identity of ‘…embraced by…’ is a case in point (Example 5.11): is it <VBN><RP> or is it <VBN><IN>? If the function word ***by*** is tagged <RP>, it falls within the subcategorisation frame of the verb and is classed as an *argument*; whereas if it is tagged <IN>, its attachment is to the ensuing noun (‘…**by** successive British **governments**…’) and its behaviour is that of an adjunct - see Merlo *et al* (2006) for recent discussion of argument/adjunct distinction for prepositions.

(1) [NP the idea] | [VP which has been embraced by]   [NP successive governments]

(2) [NP the idea] | [VP which has been embraced] | [PP by successive governments]

Example 5.11: Alternative ‘chunk’ parsing strategies for sentence fragment

The ‘blended category’ POS status (Manning and Schutze, 1999) of ***by*** in this instance is an opportunistic moment for the speaker to run with one of two different prosodies and two different parsing strategies as shown in Example 5.11. Strategy (1) is the corpus version and strategy (2) is the version created by the POS-tagger. Since both versions are *inherent in the plain text* and both are equally valid, then perhaps such ‘conflicts’ can be resolved by generating POS tagged and prosodically annotated *variants* for a given text? These parallel prosodic-syntactic realisations will then enrich the gold standard and enable more robust, i.e. ‘noise-tolerant’, evaluation of language models and contribute to our understanding of linguistic phenomena, the goal of ‘speech science’ as defined by Huckvale (2002). Moreover, the idea of including variant annotations in a gold standard has been proposed and/or adopted in other areas of computational linguistics. It is well-established that two or more linguists may disagree on the analysis/annotation of a given sample of data (Shriberg and Lof, 1991; Carletta, 1996; Bayerl and Paul, 2007); and sometimes both analyses can be legitimate. The MorphoChallenge2005 gold standard for evaluation of morphological analysis programs entered for the contest (Kurimo *et al.*, 2006) included occasional variant morphological segmentations; for example: **pitchers** can legitimately be analysed as **pitch er s,** OR **pitcher s**. Part-of-Speech taggers are normally expected to predict a single unambiguous PoS-tag for each word, but the gold standard Penn Treebank does allow for rare occasions when the Part of Speech is genuinely ambiguous (Santorini 1990, Marcus *et al.*, 1994, 82007); for example: **The duchess was entertaining last night**, the word **entertaining** is tagged **JJ|VBG** - Adjective OR Present Participle Verb. Similarly, a Multitreebank or collection of variant syntactic analyses of sentences can be used for comparative evaluation of rival parsing programs (Atwell, 1996), corpus linguists' parsing schemes (Atwell *et al.,* 2000), and unsupervised machine learning Grammatical Inference systems (van Zaanen *et al.*, 2004).

5.4. Summary

This chapter completes the extensive survey of theoretical and methodological issues pertaining to this thesis in Chapters 1, 2 and 4. One of the main themes in this chapter is prosodic variance. The sustained *discursive* analysis of prosodic variance here undertaken with close scrutiny of SEC annotations is considered to a strength in this thesis (§10.12). Aspects of this variance include the discernment of fluid syntactic categories such as particles and prepositions, and the distinction between highlighting and chunking boundaries. Consideration of empirical evidence of prosodic variance from the corpus in this chapter has influenced decision-making in thesis chapters 7 through 9. Despite being attuned to different boundary types and strengths, both as concepts and in actuality, no distinction is made in experimental work between major and minor boundaries because automatic identification of intelligible and naturalistic boundary sites in unseen text is challenging enough in itself. Moreover, researchers need to develop an awareness that however accurate their phrasing model, this achievement must be tempered by the *relativity* of correct predictions: more often than not, there are legitimate *alternative* phrasing strategies for any given sentence in English, and this subsumes the major/minor boundary divide.

Chapter 6
ProPOSEL: the ***Pro***sody and ***P***art-***o***f-***S***peech ***E***nglish ***L***exicon

6.1. Taking Stock: Motivation for ProPOSEL

One of the thematic programmes for PASCAL-2 (2008) identifies a current interest in, and trend towards, leveraging real-world knowledge to enhance performance in machine learning in a variety of application domains, including text and language processing, where previously little *a priori* knowledge has been assumed on the part of the learning mechanism (*cf.* also the CFP for IJCAI 2009 on User-Contributed Knowledge and Artificial Intelligence). The survey in Chapter 3 reveals a deficiency of *a priori* linguistic knowledge of prosody in the feature sets typically used in rule-based and data-driven phrase break models. In contrast, a competent human reader is able to project holistic linguistic insights, including projected prosody, onto text and to treat them as part of the input (Fodor, 2002). That same human reader will interpret the sound pattern signified by the orthographic form (Saussure in Chandler, 2002:18-20) when processing written language in their mother tongue. This thesis later contends that native English speakers, whether speaking, reading or writing, may use certain sound patterns as *linguistic signs* for phrase breaks. It also contends that such signs can be extracted from canonical forms in the lexicon and presented as input features for the phrase break classifier in the same way that real-world knowledge of syntax is represented in PoS tags. Moreover, such features are domain-independent and, like content-function word status, can be projected onto *any* corpus.

In addition to the inspirational *because transferable* chinks and chunks algorithm, multiple prosodic annotation tiers in the Aix-MARSEC corpus have also been revelatory, since they capture the prosody implicit in text and currently absent in learning paradigms for phrase break models. These two insights, plus an appreciation of prosodic variance gleaned from close examination of corpus evidence in Chapter 5, have informed the creation of ProPOSEL, the domain-independent lexicon and prosodic annotation tool in this thesis.

In accordance with guidelines on linguistic data management, this chapter now documents how domain knowledge from several widely-used lexical resources has been combined to create ProPOSEL, a ***pro***sody and ***p***art-***o***f-***s***peech ***E***nglish ***l***exicon of 104,049 entry groups, customised for linkage with corpora and language engineering tasks that involve the prosodic-syntactic chunking and/or analysis of text. ProPOSEL's multi-field format classifies wordforms under four variant PoS-tagging schemes mapped to default closed and open-class word categories; it offers alternative access routes for users via phonetic transcriptions, syllable counts, CV patterns and lexical stress patterns or abstract representations of rhythmic structure; and it lends itself to implementation and exploitation as a Python dictionary object, with multiple values associated with each compound lookup key. The lexicon is intended for distribution with the Natural Language ToolKit and is therefore supported by dedicated Python software and tutorial documentation (Appendix 2) for corpus-based research and NLP.

6.2. ProPOSEL: Derivation and rationale

Lexical resources have long been used in language teaching and linguistic research; they are increasingly important in computer modelling of language, development of systems and tools for machine learning, language engineering and corpus linguistics, and applications in text analytics and stylometry. A number of English lexical resources have been taken up by computing researchers; and these researchers might benefit from a single resource which combines features from these.

The 'computer usable' dictionary of wordforms known as CUV2 (Mitton, 1992) - itself derived from the Oxford Advanced Learner's Dictionary of Current English (Hornby, 1974) - has recently been updated (Pedler and Mitton, 2003). As well as increasing the number of entries from 70,646 to 72,060, *CUVPlus*[1] identifies word class via the C5 tagset, the syntactic annotation scheme used in the BNC or British National Corpus (Burnard, 2000; Leech and Smith, 2000; UCREL, 2010). This introduction of more discriminating word class information than can be captured in the eight parts-of-speech {noun; verb; adjective; preposition; pronoun; adverb; conjunction; and interjection} traditionally used in English lexica is

[1] Available from the Oxford Text Archive: http://ota.ahds.ac.uk/texts/2469.html

significant: it facilitates linkage with machine-readable corpora - like the BNC itself - used by computational linguists for a range of Natural Language Processing (NLP) tasks, though the original focus for both Mitton's and Pedler's work is computer spellchecking.

A machine-readable pronunciation lexicon is an integral part of front-end NLP modules in voice-driven applications; for example, it constitutes a natural way of giving a generic Text-to-Speech Synthesis (TTS) system both prosodic and syntactic insights into input text. For English, three such resources - originally developed for Automatic Speech Recognition (ASR) and listing words and their phonetic transcriptions - are widely used: CELEX-2 (Baayen *et al*, 1996); PRONLEX (Kingsbury *et al*, 1997); and CMU, the Carnegie-Mellon Pronouncing Dictionary (Carnegie-Mellon University, 1998). The latter is used in Edinburgh's Festival speech synthesis system (Black *et al.*, 1999; Black, 2000); and the CMU text file is included as one of the datasets in NLTK - the Natural Language ToolKit, the library of Python software, data and tutorials for teaching and research in language and computing. Finally, the need for language resources containing fine-grained grammatical, morphological and phonetic information to meet the requirements of NLP components for language engineering is well illustrated by the European-funded LC-STAR project (Hartikainen *et al.*, 2003). Wide-coverage lexica for thirteen world languages[2], including US-English, were created for flexible ASR and high-quality TTS modules in Speech-to-Speech Translation (SST) applications.

These lexical resources have been used in a wide range of linguistics and language engineering research. For example:

- OALD has been used in pronunciation prediction in speech processing systems (Davel and Barndard 2008) as well as in spelling error detection (Mitton, 1996), (Pedler 2001; 2007);

[2] The thirteen world languages represented in language-independent LC-STAR lexica are: Catalan, Finnish, German, Greek, Hebrew, Italian, Mandarin Chinese, Russian, Slovenian, Spanish, Standard Arabic, Turkish, and US-English

- CUV has been used in computational generation of limericks (Lessard and Levison, 2005), and psycholinguistic research on letter-to-sound rules in adult readers (Kessler and Treiman, 2001)
- BNC has been used for developing and evaluating dictionary and grammar resources for English language learning and language engineering (Baldwin *et al.*, 2004);
- CELEX has been used for cognitive science research on anagram analysis (Vincent *et al.*, 2006);
- PRONLEX has been used for pronunciation modelling in speech recognition systems (Hazen *et al.*, 2005);
- CMU has been used in humour research investigating humorous acronyms (Stock and Strapparava 2003), and as a guide in developing lexicons for new languages (Maskey *et al.*, 2004).

All of these applications, and many others, could benefit from an extended resource combining the information in all these lexical resources. This chapter records how entries in CUVPlus have been reorganised and supplemented with information generated from several reputable language resources (including the above) to create ProPOSEL, a purpose-built repository of linguistic concepts in accessible text file format for the target application of prosodic phrase break prediction but relevant to a range of machine learning and language engineering tasks. Since ProPOSEL is intended for open source distribution with NLTK, this paper documents the creation of this lexical resource - its derivation, content, application and associated software - in accordance with guidelines on linguistic data management in Bird *et al*, (2009, Chapter 11).

Hence, relevant documentation concerning source lexica for ProPOSEL is introduced; and entry groups in CUVPlus; CELEX-2; CMU; and ProPOSEL itself are thoroughly explored, so that potential users are aware of the challenges involved in merging information from different lexica, and appreciate the degree of variance uncovered in relation to syllable counts, vowel reduction and assignment of secondary stress in those source lexica.

The chapter also discusses the relevance of new, automatically generated and manually inspected fields in ProPOSEL to supervised machine learning tasks and particularly to phrasing algorithms; and finally covers implementation of the prosody-PoS English lexicon as a Python dictionary or associative array, and user access strategies.

6.3. Record structure in CUVPlus

Each one-line entry in CUVPlus is presented as a series of six pipe-separated fields as in Example 1.

Example 1

```
burning|0|'b3nIN|Jb%,OA%|AJ0:14,VVG:14,NN1:2|2
```

These are for: (1) orthographic form; (2) a capitalisation flag, where *zero* signals lower case and *one* upper case; (3) SAM-PA phonetic transcription; (4) word class and frequency rating from the CUV2 parent file; (5) C5 PoS tag plus enhanced CUVPlus frequency rating - rounded frequencies per million based on BNC counts; and (6) CUV2 syllable count. Fields one, three, five, and six are particularly relevant to prosodic-syntactic analysis of text and the alignment of fields one and five (i.e. orthographic form mapped to PoS category) is essential for automated processing of natural language. However, language models require a one-to-one mapping of wordform and word class. CUVPlus compacts syntactic variants for a given wordform into the same field; so securing this one-to-one mapping in ProPOSEL (Table 6.1) was a primary objective, especially since it suggests a (token, tag) tuple or pairing which would provide a unique identifier for automated dictionary lookup later (§6.7).

CUVPlus Format	Target Format for ProPOSEL

`burning|AJ0:14,VVG:14,NN1:2`	`burning|VVG:14| present participle`
	`burning|AJ0:14| adjective`
	`burning|NN1:2|  noun or gerund`

Table 6.1: Alternative formats for mapping wordform and word class information in lexica

6.3.1. Phonology fields in CUVPlus

SAM-PA phonetic transcriptions in CUVPlus mark secondary as well as primary stresses via commas and apostrophes respectively, while the last field logs number of syllables. Example 2 gives phonological information in fields one, three, and six for the wordform *objectivity*.

```
Example 2
objectivity | ,0bdZek'tIvItI | 5
```

However, while the native speaker, or advanced learner, of English can deduce that primary stress in this instance occurs on the *third* in a series of *five* syllables: *ob-jec-***ti***-vi-ty*, an automaton would need to be equipped with a sophisticated set of language-specific rules (known as the Maximal Onset Principle) to resolve problems like the following and eventually reach this conclusion:

- there is a string of fourteen characters `,0bdZek'tIvItI` which needs splitting into *n* syllables;
- slice `[0:3]` represents the first syllable `/,0b/`;
- this is because `/bdZ/` is not a legal phoneme sequence in English and therefore the syllable division must occur between `/b/` and `/dZ/`.

Fortunately, there are alternative approaches in other computer usable dictionaries which may provide stressed and syllabified phonetic transcriptions or which may represent lexical stress as a pattern of numbers, one for each syllable. Thus, the stress pattern for ,ob-jec-'**ti**-vi-ty would be 20100. The second major objective was therefore to introduce supplementary phonological information from such sources as additional fields in ProPOSEL.

6.4. English phonology in pronunciation lexica for speech technology

The LC-STAR project highlighted the need for phonetic, prosodic, and morpho-syntactic enrichment in pronunciation lexica for voice-driven applications (Hartikainen *et al*, 2003). All thirteen LC-STAR lexica conform to a language-independent specification with guidelines on coverage, syntax, and phonology, for example:

- each lexicon, defined as a set of entry group elements, includes at least 50,000 inflected common-word entries;
- generic entries classify wordforms via a basic scheme of 21 PoS with attributes common to several languages;
- phonological information takes the form of stressed and syllabified SAM-PA phonetic transcriptions.

As an aside, all function words are assigned primary stress and this is also the default setting for function words in ProPOSEL.

6.4.1. English phonology in the Carnegie-Mellon pronouncing dictionary

The CMU pronouncing dictionary restricts information for each of its 127,069 entries to: orthographic form; a counter denoting pronunciation variant; and an ARPAbet phonetic transcription - the ARPAbet being an American English subset of the International Phonetic Alphabet (IPA). Entries for the inflected form *presented*, which displays the maximum number (*i.e.* three) of American English pronunciation variants in this dictionary, are as follows (Example 3).

```
Example 3
PRESENTED 1 P R IY0 Z EH1 N T AH0 D
PRESENTED 2 P ER0 Z EH1 N T AH0 D
PRESENTED 3 P R IY0 Z EH1 N AH0 D
```

Interestingly, the phonetic transcriptions in this dictionary do not show how stress affects vowel reduction (Jurafsky and Martin, 2008); hence, the usual ARPAbet symbol for schwa /ax/ does not make an appearance (*cf.* P R IY0 Z EH1 N T **AX0** D and its counterpart in SAM-PA prI'zent@d). Also, while a stress pattern can easily be extracted from CMU's ARPAbet transcriptions, homographs cause problems because there is no syntactic information to distinguish between wordforms which have the same spelling but which belong to a different class. The lemma *present* is a case in point (Table 6.2).

CMU ENTRY	STRESS PATTERN	WORD CLASS
PRESENT 1 P R EH1 Z AH0 N T	1 0	It is not possible to automatically determine from this entry which word class this

		most common pronunciation and stress pattern belongs to. (A native speaker or advanced learner will know it's *either* a noun *or* an adjective.)
PRESENT 2 P R IY0 Z EH1 N T	0 1	Pronunciations 2 and 3 for this lemma signify that it's a verb - but how can this be automatically determined from the information given?
PRESENT 3 P ER0 Z EH1 N T	0 1	

Table 6.2: Automatic mapping of phonological and syntactic information is not enabled from CMU dictionary entries

6.4.2 English phonology in CELEX-2

There are some 160,595 English wordforms in the CELEX-2 lexical database. Phonological information is detailed: Example 4 shows an entry for *territorial* from a CELEX-based *epw* (English phonology wordforms) directory[3].

```
Example 4

90218\territorial\0\46811\2\P\"tE-r@-'t$-r71\
[CV][CV][CVV][CVVC]\[tE][r@][tO:][rI@l]\S\"tE-rI-'t$-
r71\[CV][CV][CVV][CVVC]\[tE][rI][tO:][rI@l]
```

Fields of interest are: (1) unique ID number or key; (2) orthographic form; (6) pronunciation status: primary citation form or stylistic variant of same; (7) stressed and syllabified phonetic transcription using the DISC character set; (8) CV (consonant-vowel) pattern; (9) syllabified phonetic transcription using the SAM-PA character set; (10) secondary, less common pronunciation variant.

Field seven is of particular interest. CELEX-2 provides four different character sets for phonetic transcriptions, including the DISC set which allows one-to-one mapping between character and distinct phonological segment. The DISC transcription for *territorial*: `/"tE-r@-'t$-r71/` shows the character `/7/` being used to represent a dipthong. Field seven also demonstrates that *if* the user selects a *stressed* and *syllabified* phonetic transcription, irrespective of character set, they will have effectively assigned stresses to syllables and bypassed the problems outlined in Section 3.1. A lexical stress pattern of 2010 (but *not*, unfortunately, 20100 - §Section 4.4) can also be derived for *territorial* from the DISC transcriptions in

[3] School of Computing NLP resources, University of Leeds

field seven, where `/'/` denotes primary and `/"/` secondary stress; alternatively, users familiar with Unix can use an *awk* script to compute this pattern.

6.4.3. Variance in pronunciation lexica: lexical stress patterns

The CELEX-2 database lists a number of primary (P) and secondary (S) pronunciation variants for each lemma or wordform, as in this example from the English Linguistic Guide (Burnage, 1990) for *passenger*, using stressed and syllabified SAM-PA[4] transcriptions (Table 6.3).

Variant pecking order	Pronunciation status	SAM-PA transcription
1	P	**‟p{-sIn-Dz@r***
2	P	**‟p{-sIn-Z@r***
3	S	**‟p{-s@n-dZ@r***
4	S	**‟p{-s@n-Z@r***
5	S	**‟p{-sn,-dZ@r***
6	S	**‟p{-sn,-Z@r***

Table 6.3: Primary and secondary pronunciation variants for *passenger* in CELEX-2

Despite such segmental variation, the lexical stress pattern usually remains constant for a given word of two or more syllables in a given sentence slot - at least in terms of the location of primary stress: here *passenger* is realised throughout as 100. The citation form for each entry in the CELEX-2 wordforms directory was therefore used as the main generator for lexical stress patterns in ProPOSEL.

Perhaps the notion of one abiding stress pattern for an inflectional form in English needs qualification, however; homographs (§6.5.1) are a special case and there is evidence that dictionaries differ in the assignment of secondary stress and in syllable counts. The Carnegie-Mellon pronouncing dictionary (American English) is comfortable with secondary stress on the final syllable, whereas the Oxford-Longman derived CELEX-2 and CUVPlus (British English) are not (Table 6.4).

Lexicon	**Orhographic Form**	**Phonetic Transcription**	**Stress Pattern**

[4] SAM-PA transcriptions use /”/for *primary* stress; the asterisk in “p{-sIn-Dz@r* in Example 5 denotes a linking /r/.

CELEX-2	abolished	**@-'bQ-lISt**	010
CUVPlus	abolished	**@'b0lISt**	010
CMU	abolished	**AH0 B AA1 L IH2 SH T**	012
CELEX-2	calcify	**'k{l-sI-f2**	100
CUVPlus	calcify	**'k&lsIfaI**	100
CMU	calcify	**K AE1 L S AH0 F AY2**	102
CELEX-2	finite	**'f2-n2t**	10
CUVPlus	finite	**'faInaIt**	10
CMU	finite	**F AY1 N AY2 T**	12

Table 6.4 : Secondary stress is not marked for the wordforms: *abolished*, *calcify* and *finite* in either CELEX-2 or CUVPlus, whereas Carnegie-Mellon assigns secondary stress quite readily in these cases.

6.4.4. Variance in pronunciation lexica: syllabification

When it comes to syllabification, Roger Mitton's account (Mitton, 1992) of his difficulties in deciding on syllable counts for some 3000 or so words in CUV2 is illuminating. His problems were to do with the `/@/` phoneme or schwa in dipthongs, in the middle of words and in words ending in *-ion*. He compares the sound `/aI@/` in *higher* (definitely 2 syllables) and *hire* (he is unsure). He opts for one syllable for each of: *fire/hire/wire/pier/tour* but says that '…on another day, [he] might easily have counted them as two…' He juxtaposes *gambolling* with *gambling* and gives instances of the word *champion* realised as 2 and then 3 syllables. Sometimes it's simply a case, he says, of judging whether more or less `/@/` seems most natural. Hence Mitton awards *territorial* five syllables whereas CELEX-2 only gives it four.

Mitton's experience is played out in the dictionaries themselves. Reduced vowels are included in syllable counts but sometimes disappear from phonetic transcriptions. The online version of OALD[5] (now in its seventh edition) uses the schwa in its transcription for *descendant* (3 syllables) but not for *iridescent* (4 syllables) - and the same goes for CUVPlus (Example 5).

Example 5

descendant|0|dI'send@nt|K6%|NN1:3|3

[5] Oxford Advanced Learner's Dictionary: http://www.oup.com/elt/catalogue/teachersites/oald7/?cc=global

```
iridescent|0|,IrI'desnt|OA%|AJ0:1|4
```

A quick check on the LDOCE[6] website verifies that Longman use the schwa in both transcriptions.

Finally, dictionaries are not in accord on syllabification. To give just one instance, the syllable count for *memorial* is difficult to determine because it contains a dipthong (Mitton’s old problem again). The CUV2-derived CUVPlus records a syllable count of 4 for this wordform but the CV pattern and syllabified transcription in CELEX-2 tell a different story. ProPOSEL reflects these discrepancies and this is intentional (Example 6).

```
Example 6
CUVPlus:                        memorial|0|mI'mOrI@l|K6%|NN1:16|4
CELEX-2:                          [CV][CVV][CVVC]\[m@][mO:][rI@l]
```

6.5. ProPOSEL: entry format, content, and build

ProPOSEL is a textfile of 104,049 separate entries, each comprising fifteen pipe-separated fields arranged as in Example 7.

```
Example 7
sunniest|AJS|0|'sVnIIst|Os%|AJS:0|3|100|JJS|C|JJT|JJT|
'sV-nI-Ist|'sV:1 nI:0 Ist:0|[CV][CV][VCC]
```

(1) wordform; (2) C5 tag; (3) capitalisation flag; (4) SAM-PA phonetic transcription; (5) CUV2 tag and frequency rating; (6) C5 tag and BNC frequency rating; (7) syllable count; (8) lexical stress pattern; (9) Penn Treebank tag; (10) default content or function word tag; (11) LOB tag; (12) C7 tag; (13) DISC stressed and syllabified phonetic transcription; (14) stressed and unstressed values mapped to DISC syllable transcriptions; (15) consonant-vowel pattern.

The CUVPlus ordering of fields has been preserved but shunted one place to the right to accommodate field two for duplicate C5 PoS tags stripped of their BNC frequency counts. Also, the number of entries for each wordform is proportional to

[6] Longman Dictionary of Contemporary English: http://pewebdic2.cw.idm.fr/

the number of PoS tag categories assigned to it; and it is important to note that this in turn only reflects the distribution of that wordform in the parent corpus, the BNC.

6.5.1. Phonology fields in ProPOSEL

Currently, there are eight new automatically-generated fields (fields eight to fifteen) for each entry in addition to the second field; new phonological information is held in fields eight, thirteen, and fourteen, which include alternative phonetic transcriptions (DISC) to those in the original CUVPlus file. Lexical stress patterns in field eight have been derived in the first instance from CELEX-2, and hence use the version of English known as *received pronunciation* (RP) as canonical pronunciation form; if the wordform did not appear in this database, then, where possible, the pattern was extracted from Carnegie-Mellon. It is still possible to identify which source has been used from the stress pattern entries: gaps have been preserved between digits in stress patterns derived from CMU and the slight difference in presentation is deliberate (Table 6.5).

betting	VVG	10	Lexical stress pattern generated from **CELEX-2**
betting	AJ0	10	
bettor	NN1	1 0	There are gaps between digits in the lexical stress patterns derived from **Carnegie-Mellon**
bettors	NN2	1 0	
betty	NP0	1 0	

Table 6.5: Extracts from 5 adjacent wordform entries in ProPOSEL

This exercise currently still leaves 7,816 out of the 104,049 entries in the prosody lexicon with ‘No value’ recorded in the stress pattern field - typical examples being things like: *a bit*; *‘tween-decks*; *bletchley*; and *blighty*. Similarly, there must be a considerable number of unused stress patterns from CELEX-2 and Carnegie-Mellon which did not find a matching entry in CUVPlus and hence ProPOSEL.

6.5.2. Homographs

One of the most fascinating aspects of this work has turned out to be homographs: words with one spelling but two different pronunciations and two distinct meanings and/or usages: *bass*, *present* and *wound* are classic examples. A comprehensive list of some 550 homographs - most of which are derived from none other than Mitton’s computer-usable dictionary - is available from Higgins (2010)[7].

One field of particular interest to our research into automatic phrase break prediction is lexical stress pattern, where the rhythmic structure of wordforms is represented symbolically as a string of numbers. For some homographs, this lexical stress pattern can fluctuate depending on part-of-speech category and meaning. Rhythmic structure for the homograph *present* is inverted when it functions as a verb, for example, as shown in fields one, two, four, seven, eight and ten for all its entries in ProPOSEL (Example 8).

```
Example 8
present | AJ0 | ’preznt | 2 | 10 | C |
present | NN1 | ’preznt | 2 | 10 | C |
present | VVI | prI’zent | 2 | 01 | C |
present | VVB | prI’zent | 2 | 01 | C |
```

All homographs have been checked and where necessary, entered manually when compiling the prosody-PoS English lexicon.

6.5.3. Word class annotations in ProPOSEL

So far, three rival PoS-tagging schemes have been included in ProPOSEL in addition to C5 for linkage with different corpora: Penn Treebank (Marcus *et al.*, 1994); LOB (Johansson *et al.*, 1986); and C7 (*cf.* UCREL, 2010). After much reflection and experimentation, a Penn Treebank ⇔ C5 mapping[8] was achieved, based on - but not identical to - Naber’s mapping of same, available online (Naber, 2003). One advantage of including this annotation scheme is that there are suggested default mappings of Penn Treebank to content-function word and punctuation (CFP) tags (Busser *et al.*, 2001; Bell, 2005); and therefore default settings were automatically generated from the Penn Treebank field for the CFP field in ProPOSEL (§5.2). This lexicon object of ten fields constituted the first prototype for

[7] Formerly available through the British Library Net internet service until it was discontinued on 31 March 2007

[8] See Appendix 1

ProPOSEL (*cf*. Figure 1). For the additional syntactic fields, a C5 ⇔ C7 mapping[9] was available from the UCREL website and a suggested mapping of C5 ⇔ LOB was documented in Pedler's PhD thesis (Pedler, 2007); all mappings of C5 to variant PoS-tagging schemes in ProPOSEL are cross-referenced against these sources.

The challenges of converting between different tagsets have been well documented (*cf*. Atwell *et al.*, 1994; Atwell *et al.*, 2000; Atwell, 2008). The exercise recently undertaken reveals that syntactic information is lost both ways even when mapping between C5 and Penn, both relatively lean tagsets; and this problem is compounded with richer tagsets like LOB and C7. One-to-many mappings uncover 'indelicate' areas of each tagset. For example, Penn uses just one tag **IN** for prepositions and subordinating conjunctions (*cf*. AMALGAM, 2010) whereas C5 deploys a range of tags: **PRF** (the preposition *of*); **PRP** (any other preposition); **CJS** (subordinating conjunction); **CJT** (the conjunction *that*). Similarly, Penn has separate tags for *whose* (WP$) and for pre-determiners (PDT). Since prosodic phrasing is sensitive to clause markers (Koehn *et al*, 2000) and since pre-determiners (e.g. *all*, *quite*, and *this*) are good candidates for pitch accent reinforcement, prosodic-syntactic clues are thus potentially compromised if such differentiations are subsumed under one tag.

Example 9 shows a somewhat daunting array of one-to-many mappings from C5 (field two) to equivalent sets of PoS-tags in Penn (field nine), LOB (field eleven), and C7 (field twelve) for just one wordform *ably* in ProPOSEL's textfile.

Example 9

```
ably|AV0|RB,RBR,RBS|QL,QLP,RB,RI,RBR,RBT,RN|
BCL,RA,REX,RG,RR,RL,RGR,RGT,RRR,RRT,RT
```

The BNC tagset is relatively sparse and uses {AV0} as a catch-all tag for all kinds of adverbs bar particles {AVP} and wh-adverbs {AVQ}. This then generates one-to-many mappings from C5 to other syntactic annotation schemes where subsets of the adverbial category proliferate (*cf*. Nancarrow and Atwell, 2007). Variant PoS-tag fields in ProPOSEL list all subsets of the parent category which matches the C5 tag

[9] UCREL C5 > C7 Mapping:
http://www.comp.lancs.ac.uk/ucrel/claws/mapC7toC5.txt

for each entry group, and for this reason, the lexicon only lends itself to *generation* of PoS candidates for raw text via the C5 scheme. It would be possible, however, to generate a parallel syntactic analysis of a corpus like MARSEC where the original LOB annotations were mapped to C5 via automated look-up in ProPOSEL. Further instances of one-to-many mappings are provided in Table 6.6 to demonstrate how enclitics and Saxon genitives are presented in CUVPlus and handled during lexicon build. ProPOSEL is intended for open source distribution with NLTK and is supported by a toolkit of Python software, plus an explanatory tutorial, for negotiating such complexity, including a section on preparing the textfile for NLP (Appendix 2).

Wordform & C5 Tag	Penn, LOB and C7 Fields
he'd|PNP+VM0	PRP,MD|PP3AS,PP3O,PP3OS,PP$$,PP1A,PP1AS,PP1O,PP1OS,PP2,PP3,PP3A**+MD**|PPGE,PPIS1,PPIS2,PPIO1,PPIO2,PPY,PPH1,PPHS1,PPHS2,PPHO1,PPHO2+**VM**,**VMK**
lloyd's|NP0+POS	NNP+NNPS**+POS**|NP,NPL,NPLS,NPS,NPT,NPTS**+$**|NPD1,NPD2,NPM1,NNL1,NNL2,NP,NP1,NP2,NNA,NNB**+GE**
beyond|AV0	RB,RBR,RBS|QL,QLP,RB,RI,RBR,RBT,RN|BCL,RA,REX,RG,RR,RL,RGR,RGT,RRR,RRT,RT

Table 6.6: Examples of one-to-many mappings in selected fields from raw text file entries in ProPOSEL for enclitics, Saxon genitives and adverbs.

6.5.4. Summary of lexicon build

Building ProPOSEL was accomplished in two main stages, recorded in the flowchart summary in Figure 6.1. The first stage involved:

- generating lexical stress patterns from CELEX-2 and CMU;
- mapping these patterns to wordform entries in CUVPlus;
- generating one-to-one mappings of wordform to word class in the emergent lexicon;
- mapping content-function word categories to Penn Treebank tags;
- mapping the above to word class (C5 tags) in the first prototype lexicon.

The prototype prosody and PoS English lexicon thus had entries with ten fields: the original six fields in CUVPlus and additional fields for C5 PoS tags, lexical stress patterns, Penn Treebank tags, and content-function word defaults. The second stage involved:

- supplementing entries in the prototype lexicon object with C5 > LOB, and C5 > C7 mappings in two separate steps;

- revisiting CELEX-2 to capture DISC syllabified transcriptions and CV patterns, and to award each syllable a stress value of {0, 1, or 2};
- appending these as three extra fields for each entry in the lexicon;
- manually inspecting and correcting all homographs;
- diagnostic testing of all fields and simultaneous development of user access code for ProPOSEL.

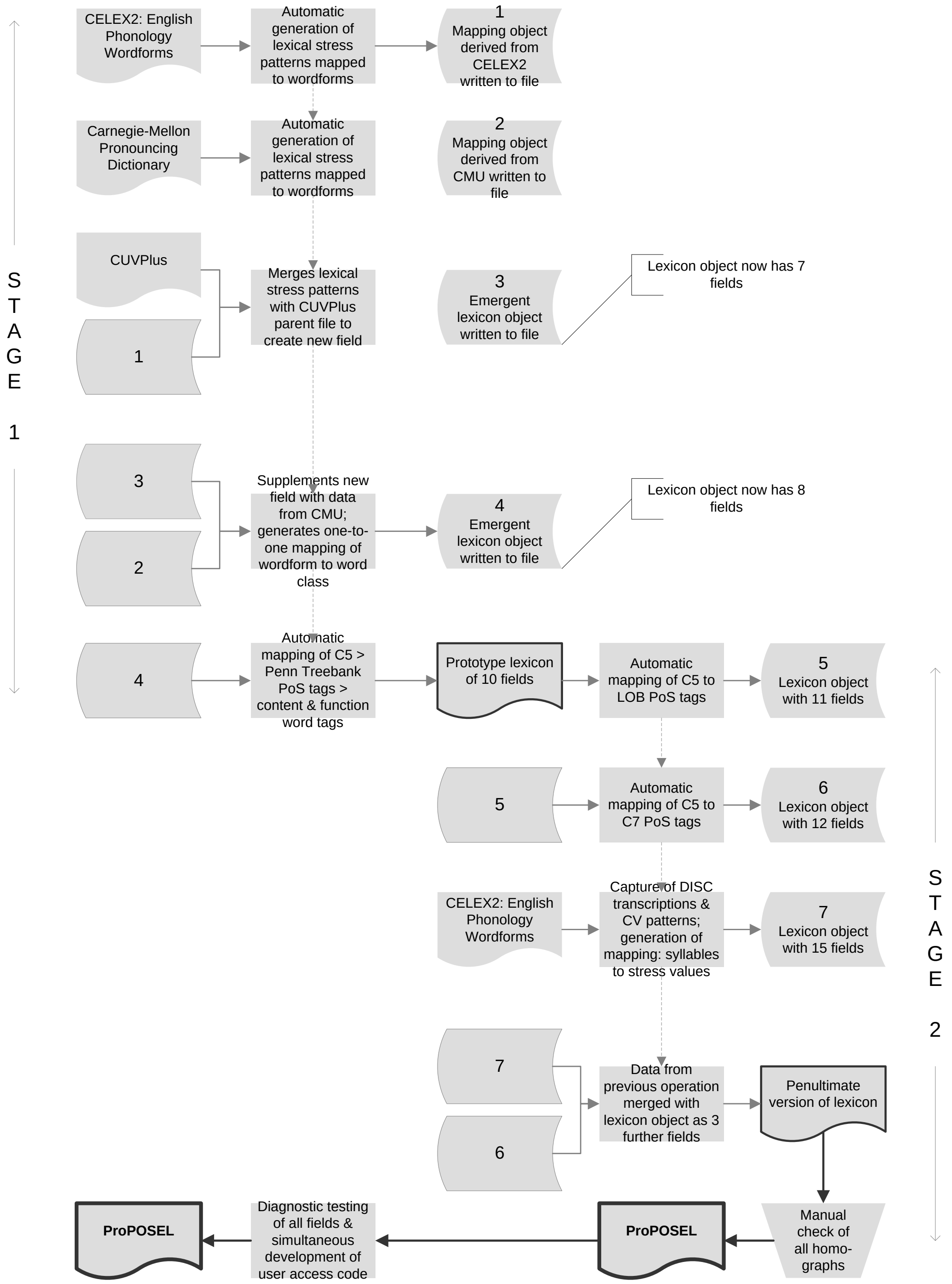

CELEX2: English Phonology Wordforms
Automatic generation of lexical stress patterns mapped to wordforms
1 Mapping object derived from CELEX2 written to file
Carnegie-Mellon Pronouncing Dictionary
Automatic generation of lexical stress patterns mapped to wordforms
2 Mapping object derived from CMU written to file
CUVPlus
1
Merges lexical stress patterns with CUVPlus parent file to create new field
3 Emergent lexicon object written to file
Lexicon object now has 7 fields
STAGE 1
3
2
Supplements new field with data from CMU; generates one-to-one mapping of wordform to word class
4 Emergent lexicon object written to file
Lexicon object now has 8 fields
4
Automatic mapping of C5 > Penn Treebank PoS tags > content & function word tags
Prototype lexicon of 10 fields
Automatic mapping of C5 to LOB PoS tags
5 Lexicon object with 11 fields
5
Automatic mapping of C5 to C7 PoS tags
6 Lexicon object with 12 fields
CELEX2: English Phonology Wordforms
Capture of DISC transcriptions & CV patterns; generation of mapping: syllables to stress values
7 Lexicon object with 15 fields
STAGE 2
7
6
Data from previous operation merged with lexicon object as 3 further fields
Penultimate version of lexicon
Manual check of all homo-graphs
ProPOSEL
Diagnostic testing of all fields & simultaneous development of user access code
ProPOSEL

Figure 5.1: Flowchart summary of ProPOSEL Build

Figure 6.1: Flowchart summary of ProPOSEL build

6.6. Dictionary-derived features for machine learning of prosodic-syntactic chunking

As previously stated, the purpose of this work is to integrate information from different dictionaries into one lexicon, customised for language engineering tasks which involve the prosodic-syntactic chunking of text. One such task is automated phrase break prediction: the identification of potential pauses in *text* which reflect the way in which a native speaker might process or chunk that same text as speech. This is treated as a classification task in supervised machine learning, where junctures (whitespaces) between words in the input text are classified as either breaks (the minority class) or non-breaks. The machine learner is trained on boundary-annotated text, the hand-labelled speech corpus or "gold standard", and then tested on an unseen reference dataset from the same corpus, *minus* the boundaries, to see how many junctures have been correctly classified.

6.6.1. The importance of PoS tags

Training and testing language models on a "gold standard" corpus which exemplifies the rules and structures to be learned is an approach widely used in NLP (e.g. in PoS-tagging, parsing, and semantic representation such as thematic role labelling). This approach depends on PoS-tagged text; the sentence fragment below (Example 10) is taken from Section A09 (informal mid-1980s BBC radio news commentary) of MARSEC, the Machine Readable Spoken English Corpus (Roach *et al*, 1993) and shows syntactic annotations from the LOB tagset and boundary annotations in the form of pipe symbols: (|) for tone unit boundary and (||) for pause (Roach, 2000).

```
Example 10
internal/JJ leaders/NNS | who've/WP+HV come/VBN together/RB to/TO
form/VB a/AT new/JJ government/NN | to/TO get/VB on/RP with/IN
it/PP3 ||
```

Therefore, a dictionary for NLP and linkage with corpora needs discriminating word class information in the form of PoS tags rather than categories based on the traditional 8 parts-of-speech; CELEX-2, for instance, only uses 9 categories to classify English lemmas (Burnage, 1990): {Noun, Adjective, Quantifier/Numeral, Verb, Pronoun, Adverb, Preposition, Conjunction, Interjection}. The LOB tagset captures fine-grained distinctions - *on* and *with* are tagged as particle (RP) and preposition (IN) respectively in the string *get on with* in Example 11 - and offers a choice of tags for the same word depending on its function or sentence-slot (Atwell, 2008). This is important for prosody - *cf.* the discussion in Section 4.3.6 of prepositional phrase attachment and its implications for prosodic-syntactic chunking. The C5 tagset used in CUVPlus, while sparser than LOB, retains this discriminatory characteristic; as noted, it has a separate tag for *of* (PRF) as distinct from other prepositions (PRP), which may emerge as a useful refinement for phrase break prediction.

6.6.2. CFP status

Phrase break classifiers have been trained on additional text-based features besides PoS tags. The CFP status of a token - is it a *content* word (e.g. nouns or adjectives) or *function* word (e.g. prepositions or articles) or *punctuation* mark? - has proved to be a very effective attribute in both deterministic and probabilistic models (Liberman and Church, 1992; Busser *et al*, 2001) and therefore, a default content-word/function-word tag is assigned to each entry in the prosody lexicon in field ten. It is anticipated that further research will suggest modifications to this default status when the CFP attribute interacts with other text-based features. For example, the word *against* is a function word but it is also bi-syllabic and likely to carry word stress - different, therefore, from function words that ‘disappear’ prosodically due to vowel reduction. The second entry for *can* in the Carnegie-Mellon pronouncing dictionary indicates this is what happens when, presumably, *can* is being a modal auxiliary (Example 11).

Example 11

CAN 1 K AE1 N (full vowel probably signifies noun)

CAN 2 K AH0 N (no schwa, no word class but looks like vowel reduction in *can* the verb)

6.6.3. Syllable count and lexical stress

Syllable counts have already been used in phrase break models for English (Atterer and Klein, 2002; Schmid and Atterer, 2004). This rather assumes uniformity in terms of duration of syllables whereas we know that in connected speech, an indefinite number of unstressed syllables are packed into the gap between one *stress pulse* (Mortimer, 1985) and another, English being a *stress-timed* language (Hirst, 2009). A lexical stress pattern, capturing both syllabification and stress distribution (rhythmic structure) in a simple abstract form, has therefore been included for each entry in the prosody-PoS lexicon because of its potential as a classificatory feature in the machine learning task of phrase break prediction. This intimation is further supported by the presence of rhythmic annotation tiers in the Aix-MARSEC corpus project (Hirst *et al.*, 2000; Auran *et al.*, 2004), with its focus on speech synthesis applications and the theoretical modelling (acoustic, phonetic and phonological) of intonation and speech prosody.

6.6.4. Prior knowledge for machine learning

One of the thematic programmes for PASCAL[10] (2008) identifies a current interest in, and trend towards, leveraging *a priori* knowledge to enhance performance in machine learning in a variety of application domains, including text and language processing. is customised for corpus-based research; and specifically, for populating raw training data (*i.e.* the tokenized corpus text) with *a priori* knowledge gathered and cross-referenced from widely-used lexica. Predicting phrase boundaries at the prosody-syntax interface is a notoriously complex task for machine learning because of the inherent variance of prosody (*cf.* Taylor and Black, 1998; Atterer and Klein, 2002; Chapter 5). Planned research into the phrase break prediction task will attempt to incorporate a dictionary-derived feature such as lexical stress pattern (field eight in ProPOSEL) into a data-driven model to explore this interface more fully; although using just the raw pattern would entail one hundred and twenty-four separate values for this feature (*i.e.* the set of lexical stress patterns in the lexicon).

6.7. Implementing ProPOSEL as a Python dictionary

A lexicon designed for linkage with corpora - speech corpora in this case - needs word class information in the form of PoS tags. It also needs wordforms and inflected forms rather than lemmas. CELEX-2, for example, lists *wake* and *waken* in its directory for English lemmas but not *woke* and *woken*; the latter appear in its directory for English wordforms instead. If we annotate these words using the LOB tagset, we get: `wake/NN wake/VB waken/VB woke/VBD woken/VBN`. The form and function of each word token in the corpus is defined by its PoS tag and therefore (token, tag) can act as a unique identifier for dictionary lookup.

The Python programming language has a dictionary mapping object with entries in the form of (key, value) pairs. Each key must be unique and immutable (e.g. a *string* or *tuple*), while the values can be any type (e.g. a *list*). This syntax can

[10] PASCAL: Pattern Analysis, Statistical Modelling, and Computational Learning: http://www.cs.man.ac.uk/~neill/thematic08.html

be exploited by transforming ProPOSEL into a Python dictionary with compound keys (wordform and C5 tag) and multiple values[11] in the form of lists of tokens from chosen fields for a given entry. In Example 12, the syntax { } denotes a dictionary data structure in Python, where keys and values are separated by a colon, and where each individual entry exhibits the structure: **(** ( ), [ ] **)** - entry groups in tuple format; immutable dictionary keys in tuple format; values associated with those keys held in a list.

```
Example 12
{((‘cascaded’,  ‘VVD’):[‘k&'skeIdId’,  ‘3’,  ‘010’,  ‘C’]),
((‘cascaded’,  ‘VVN’):[‘k&'skeIdId’,  ‘3’,  ‘010’,  ‘C’]),
((‘cascading’, ‘VVG’): [‘3’, ‘010’, ‘C’]), ((‘cascading’, ‘AJO’):
[‘3’, ‘010’, ‘C’])}
```

Thus incoming corpus text, also in the form of (token, tag) tuples, can be matched against ProPOSEL’s keys; and thus intersection enables corpus text to accumulate additional prosodic and syntactic annotations which constitute potential features for machine learning tasks.

6.7.1. Dictionary lookup when input text is not tagged with C5

The aforementioned lookup mechanism is relatively straightforward for corpora tagged with C5. A possible solution for input text annotated with an alternative PoS-tagging scheme is to find a match for (token, tag) in more than one field: field one (orthographic form) and then either field nine (Penn), or eleven (LOB), or twelve (C7). The preferred solution appends a C5 tag to each item in the input text such that lookup can proceed in the normal way. This negotiates problems caused by one-to-many mappings, enclitics and Saxon genitives, aptly illustrated by the raw textfile entry in Example 13, where the orthographic form *'twould* is an enclitic. The modern-day equivalent *it’d* would exhibit the same complexity but does not appear in CUVPlus or ProPOSEL.

```
Example 13
'twould|PNP+VM0|PRP,MD|F|PP3AS,PP3O,PP3OS,PP$$,PP1A,PP1AS,PP1O,
PP1OS,PP2,PP3,PP3A+MD|PPGE,PPIS1,PPIS2,PPIO1,PPIO2,PPY,PPH1,PPHS1,
PPHS2,PPHO1,PPHO2+VM,VMK
```

[11] See, for example, the recipe by M. Chermside in Martelli *et al.* (2005: 173)

In this example, the C5 tag only finds a one-to-one match with Penn. The forest of one-to-many mappings from *personal pronouns* (PNP) in C5 to equivalent tags in LOB and C7 is particularly dense; and C7 also has two tags for modals: VM for auxiliaries (*can*, *will*, *would* etc.) and VMK for catenatives (*ought*, *used*).

User access code for ProPOSEL includes functions for creating and tokenizing encliticised forms in all PoS fields such that 'PNP+VM0' in C5 would be mapped to 'PP3AS+MD', 'PP3O+MD' et cetera in LOB, and similarly in C7. Preserving enclitics is considered to be important for prosody; *it's* and *it is* are syntactically equivalent but are different in terms of beats. Interestingly, the archaic encliticised form *'twould* in Example 14 has a different rhythmic structure from the more familiar *it'd*.

6.7.2. Navigational software tools for ProPOSEL

ProPOSEL is supported by a toolkit of software solutions compatible with NLTK and an explanatory tutorial with sections on: preparing the textfile for NLP; mapping variant syntactic information (with subsidiary sections on handling enclitics, Saxon genitives and one-to-many mappings); implementing ProPOSEL as a Python dictionary; annotating PoS-tagged corpora with domain knowledge of phonology; and customising searches via multiple criteria.

Phonology fields in ProPOSEL constitute a range of access routes for users and enable lookup via sound, syllables, and rhythmic structure as alternatives to orthographic form. It is also possible to read in the textfile as a nested structure and perform filtered searches on particular fields or field combinations as user-defined subsets of the lexicon.

6.8. Filtered searches and having fun with ProPOSEL

The previous section demonstrated how fine-grained grammatical distinctions in the PoS tag field(s) in ProPOSEL are integral to linkage with corpora. It also demonstrated how an electronic dictionary in the form of a simple text file can be reconceived and reconstituted as a computational data structure known as an associative memory or array.

It is not always necessary to transform ProPOSEL into a Python dictionary, however. Users can also read in the lexicon textfile, apply Python's splitlines() method to process the text as a list of lines, and then apply the split() method, with the *pipe* field separator as argument, to tokenize each field. Listing 6.1 presents this much more succinctly.

```
lexicon = open('filepath', 'rU').read()
lexicon = lexicon.splitlines()
lexicon = [line.split('|') for line in lexicon]
```

Listing 6.1: Reading in ProPOSEL as a nested structure

Users can then perform a search on a defined subset of the lexicon. For example, users may wish to retrieve all entries with seven syllables from the lexicon. As well as returning items like: *industrialisation*, *operating-theatre*, and *radioactivity*, Listing 6.2 discovers the rather intriguing *sir roger de coverley*!

```
for index in lexicon:
if index[6] == '7': # look in the subset
print index[0] # return word form(s)
```

Listing 6.2: Searching a subset of the lexicon

Another illustration would be finding words which rhyme. If we wanted to find all the words which rhyme with *corpus* in the lexicon, we could search field (4), for example, the SAM-PA phonetic transcriptions, for similar strings to `/'kOp@s/`. One way of doing this would be to compile a regular expression, substituting the metacharacter **.** for the '*c*' in *corpus* and then seek a match in the SAM-PA field. We might also look for minimal pairs, replacing the phoneme `/s/` with the phoneme `/z/` as in `/'.Op@z/`. Retaining the apostrophe as diacritic for primary stress before the wildcard here imitates the lexical stress pattern for *corpus* and is part of the rhyme. It transpires there is only one candidate which rhymes with *corpus* in the lexicon and two half rhymes. Listing 6.3 gives us *porpoise* `/'pOp@s/` and then *paupers* `/'pOp@z/` and *torpors* `/'tOp@z/`.

```
p1 = re.compile("'.Op@s")
p2 = re.compile("'.Op@z")
sampa = [index[3] for index in lexicon]
rhymes1 = p1.findall(' '.join(sampa))
rhymes2 = p2.findall(' '.join(sampa))
```

Listing 6.3: Using regular expressions to retrieve bi-syllabic words with primary stress on the first syllable that rhyme with *corpus*. Note that Python lists start at index 0, hence in Listing 6.3, the SAM-PA field is at position [3] in the inner list of tokenized list fields for each entry.

Two well established phonetic transcription schemes are also represented in ProPOSEL: the original SAM-PA transcriptions in field 4 and DISC stressed and syllabified transcriptions in fields 13 and 14 which, unlike SAM-PA and the International Phonetic Alphabet (IPA), use a single character to represent dipthongs: `/p8R/` for *pair*, for example.

Phonology fields in ProPOSEL constitute a range of access routes for users. As an illustration, a search for like candidates to the verb *obliterate* might focus on structure and sound: verbs of 4 syllables (field 7), with primary stress on the *second* syllable (field 8), and with vowel reduction on the *first* syllable (a choice of phonetic transcription fields). This filter retrieves sixty-two candidates - most but not all of them end in `/eIt/` - including these in Table 6.7.

```
('affiliate', "@'fIlIeIt")
('corroborate', "k@'r0b@reIt")
('manipulate', "m@'nIpjUleIt")
('originate', "@'rIdZIneIt")
('perpetuate', "p@'petSUeIt")
('subordinate', "s@'bOdIneIt")
('vociferate', "v@'sIf@reIt")
```

Table 6.7: Sample of 7 candidate verbs retrieved which share phonological features with the template verb: *obliterate*

6.9. Chapter summary

This chapter describes a purpose-built prosody and PoS English lexicon to integrate and leverage domain knowledge from several well-established language resources for corpus-based research in speech synthesis and related fields. It is planned to make this lexicon, and the accompanying software and tutorial, freely available under the auspices of open source projects such as the Python-based Natural Language Toolkit and/or the Aix-MARSEC corpus project. The incorporation of different tagsets - currently C5, Penn, LOB, and C7 - facilitates linkage with some of the main English language corpora used in speech and language processing.

As well as arguing the case for word class identification via PoS tags in electronic dictionaries, this chapter has explored variations in syllabification and levels of prominence in the treatment of vowels and in phonetic transcriptions in English lexica. The chapter also interprets lexical stress as a potential text-based feature for supervised machine learning.

It is further suggested how a computer-usable and human readable dictionary text file can be reconceived and dynamically reconstituted as an associative array - a Python dictionary - where the recommended access strategy is via compound keys (wordform and C5 tag) which uniquely identify each lexical entry. Users can also manipulate the text file to perform filtered searches on subsets of the lexicon and access wordforms via sound, syllables and rhythmic structure.

The contributing lexical resources which formed the basis of ProPOSEL – OALD, CUV, BNC, CELEX, PRONLEX, CMU, Penn Treebank, and LOB – have each been used in a variety of research projects covering psycholinguistics, language engineering and corpus linguistics. ProPOSEL combines the lexical information from all these resources, and so should be applicable in all these research areas, and many more.

Chapter 7

Experimentation with ProPOSEL (Part 1): Derivation and Significance Testing of Non-traditional *Prosodic* Phrase Break Features in a Corpus of Seventeenth Century Blank Verse

‘…To be, or not to be: that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them? To die: to sleep…’

Hamlet Act 3, Scene 1; lines 56-60

7.1. Discussion of background (literary) terminology for prosody

The opening lines of Hamlet’s famous soliloquy contain an unpunctuated section (lines 57-58) which has given rise to much syntactic controversy; the Arden Shakespeare’s *Hamlet* (Jenkins, 2003:277) notes contradictory interpretations of prepositional phrase attachment by two critics for ‘…*in the mind*…’: is it an adverbial or adjectival prepositional phrase, modifying *suffer* or *nobler* respectively? In the first case, the phrase will initiate a new syntactic chunk and the *caesura*, or main prosodic mid-line break (*cf*. Knowles, 1987:182) will fall after *nobler*; in the second case, the phrase will complete an information unit and *tone group* (*cf*. McCarthy, 1991:99) and the caesura will fall immediately after the fourth beat in the line, after *mind*.

The above analysis introduces several important terms for discussing the prosody-syntax interface in Shakespearian blank verse and in English generally. It mentions *chunking*, that is the use of pitch accents and pauses to signal boundaries between meaningful clusters of words which have both prosodic and syntactic coherence: tone groups (prosodic units) and function word groups (syntactic units). It also mentions the *caesura* which, sometimes openly and sometimes unobtrusively and even *negligibly*, divides each line in iambic pentameter (blank verse metre) into one of the following beat patterns: {2-3; 3-2; 1-4; 4-1}. Finally, it mentions rhythm or *beats*, lexical stresses and salient accents assigned to specific syllables by the poet himself, his actors, and his readers, but again, open to interpretation. Knowles, for example (Knowles, 1987:159;181) suggests that the number of accents in a pentameter may vary. While this is so, there will always be five beats (or stresses) and, like salient accents, these beats will always fall on stress prone syllables in any given word. Thus, in the above extract from Hamlet's soliloquy, the phrase '…outRAgeous FORtune…' carries two beats as marked, but only one of them will be made salient by changes in pitch. The lexical stress patterns in '…outRAgeous FORtune…' are canonical and may be represented abstractly as a series of numbers 010 10 from the set: {0 unstressed, or weakly stressed syllable; 1 primary stress; 2 secondary stress}.

So far, we have discussed these famous lines without explicit recognition of the feeling self – but that is not to say it is absent. However, before we can interpret and respond to these lines, we need to chunk them into meaningful groups of words, to parse them in effect; and this entails making decisions about prosody. So where *is* the caesura in line 57?

'…*Whether 'tis nobler in the mind to suffer…*'

One possible rendition is to put it after *mind* because it is rhythmically more pleasing. A pause after *nobler* would insert a partition between two accent groups *'tis nobler* and *in the mind* which ignores the invitation to run them together (*nobler* ↺ *in*) via a linking *r* (Mortimer, 1985:46). This, in turn, encourages salience to gather on the word *suffer* instead of *mind*. A holistic phrasing for lines 56-60 might be as follows (with major phrase boundaries marked via || and salient items in **bold**). However, it is a matter of individual choice, as long as the phrasing and accenting make sense.

'…To **be**, or **not** to be: || **that** is the question: ||
Whether 'tis **no**bler in the mind || to **su**ffer
The slings and **a**rrows of out**ra**geous fortune, ||
Or to **take arms** against a **sea** of troubles, ||
And by op**po**sing **end** them? || To **die**: to **sleep**…'

There is one further term requiring explanation and that is *enjambement* or *run-on* lines, an instance being lines 57-58 in the above rendition, where prosodic-syntactic chunking does not correspond to the metrical lines. The absence of punctuation at the end of line 57 is a clue, used elsewhere in this speech and in Shakespearian verse in general, prompting the reader to ignore the line break and process the phrase '…*to suffer the slings and arrows*…' as one chunk. Knowles (1987:138) advocates a middle way:

> '…*When you read…verse aloud, you can read according to the metre, or… according to the sense. Or, more likely, you will do something in between, trying not to lose either the metre or the sense entirely*…'

The important point to note is that prosodic phrases in blank verse occur *within* lines and *between* lines; and that the use of enjambement and the shifting location of caesuras '…helps to create an illusion of natural speech…' (V&A, 2010).

7.2. Legitimising variant phrasing in different editions of the same text

In supervised machine learning, the task of predicting prosodic phrase breaks in text which mimic human phrasing equates to classifying junctures or whitespaces between words as either breaks (the minority class) or non-breaks. Given the interdependence of prosody and syntax, the language model is invariably trained on part-of-speech (PoS) contexts in which boundaries are likely to occur. These contexts are defined by a *gold standard,* an annotated speech corpus, processed as a list of tokens comprising PoS tags (syntactic annotation) and human-labelled boundaries. Once the model has been trained, it is tested on an unseen reference dataset *minus the boundaries* from the same corpus, and evaluated by seeing how many of the original boundary locations have been recaptured or predicted by the model.

Evaluation against a human-labelled gold standard is a tried and tested method in computational linguistics. When applied to prosody, however, this procedure is problematic because prosody is inherently variable and the corpus '…*only represents* one *out of the space of all acceptable phrasings*…' (Atterer and Klein, 2002). Thus predictions made by a language model which do not match the corpus would be classed as insertion or deletion *errors* irrespective of their potential validity as alternative phrasings.

We have already examined one instance of prosodic variance in five lines from *Hamlet* in the previous section. The next step is to compare prosodic phrasing in eTexts of antique and modern versions of Hamlet's soliloquy, where punctuation is assumed to be a gold standard boundary marker. Tags for punctuation are included in PoS-tagging schemes and exploited in automatic phrase break prediction because of their high correlation with boundaries. Again, for the purposes of illustration, only an excerpt from the soliloquy will be used: lines 60-65.

Table 7.1 shows phrasing variance, as signified by the presence or absence of punctuation, between a modern version of *Hamlet* Act 3, Scene 1, lines 60-65 and Project Gutenberg's First Folio edition of same; both corpora are distributed with NLTK. The extracts exhibit a high degree of verse-sentence divergence; punctuation and caesuras demarcate prosodic-syntactic boundaries between independent clauses which span three lines and '*…overflow [run-on lines] is unimpeded…*' (Langworthy, 1931). It is also worth pointing out that the *placing* of punctuation (our boundary measurement) in the modern version of this extract matches exactly that of the Arden Shakespeare edition (Jenkins, 2003) although there are differences in types of punctuation mark used.

MODERN VERSION	FIRST FOLIO VERSION
…To die: to sleep; No more; and by a sleep to say we end The heart-ache and the thousand natural shocks That flesh is heir to, 'tis a consummation Devoutly to be wish'd. To die, to sleep; To sleep: perchance to dream:	…to dye, to sleepe No more; and by a sleepe, to say we end The Heart-ake, and the thousand Naturall shockes That Flesh is heyre too? 'Tis a consummation Deuoutly to be wish'd. To dye to sleepe, To sleepe, perchance to Dreame;
LINEAR PROSE-STYLE PHRASING REPRESENTATION	
…To die | to sleep | no more | and by a sleep to say we end the heart-ache and the thousand natural shocks that flesh is heir to | 'tis a consummation	…to dye | to sleepe no more | and by a sleepe | to say we end the Heart-ake | and the thousand Naturall shockes that Flesh is heyre too | 'Tis a consummation

devoutly to be wish'd | To die | to sleep | to sleep | perchance to dream |…	Deuoutly to be wish'd |To dye to sleepe | to sleepe | perchance to Dreame |…

Table 7.1: Prosodic variance captured via punctuation in different editions of Shakespeare

It seems fair to say that the First Folio extract gives the speaker-reader clearer directions for prosodic-syntactic phrasing over lines 61-63. At the same time, it introduces further uncertainty: Wilson-Knight (2001:346) calls this phraseology '…*at once inclusive and enigmatic*…' The next section discusses part-of-speech tagging; but how should we tag the phrase *No more* at the beginning of line 61? Does it mean *enough*; or *no longer*? Or is it an article-pronoun combination? And what does *to* mean in *die to sleepe* (line 64)? Is it infinitival or purposeful: *in order to* or *so as to*; or is it a verb particle? A sensitive rendition would resolve these questions, but only temporarily.

7.3. Inspecting CFP boundaries in an annotated extract from *Hamlet*

CFP rules are widely used for assigning phrase breaks in text-to-speech synthesis (TTS) applications (Abney, 1994; Knill, 2009); such rules interpret punctuation as a boundary marker and also insert boundaries between designated open-class or content words (the chunks) and closed-class or function words (the chinks). It is possible to use a stoplist (e.g. the stopwords corpus (reference) distributed as part of NLTK) as a means of filtering plain text for these grammatical words; but the usual method is to annotate the text with part-of-speech tags and thus identify lexical words {nouns; verbs; adjectives; adverbs} as distinct from all the rest.

UCREL offers a web-based free trial PoS-tagging service and this was used for syntactic annotation of Hamlet's soliloquy in this demonstration because there is an option to select the C5 tagset, simplifying the lookup process and subsequent annotation with additional features via ProPOSEL. Four fields in ProPOSEL, denoting: word form; C5 PoS tag; default content-function word tag; and stressed and unstressed values mapped to DISC syllable transcriptions were selected for this study. An instance in the output text file displaying this information would be: `patient|AJ0|C|'p1:1 SHt:0`.

7.3.1. Annotating the text with ProPOSEL

The procedure for annotating prosodic-syntactic boundaries in Hamlet's soliloquy (*cf.* Listing 7.1) was as follows:

1. C5 PoS tags for a modern English version of the text *using punctuation which accorded with Project Gutenberg's First Folio edition* were obtained via UCREL's free trial service. The Early Modern English (EModE) text had been manually pre-processed to (i) restore past tense verb forms *wish'd* → *wished*; (ii) incorporate the following transformations: *perchance* → *perhaps*; *'tis* → *it's*[12]; *aye* → *yes*; and (iii) insert the word *these* before *fardles*, as in the First Folio edition. Readers are referred to the VARD or spelling Variant Detector tool in Rayson *et al* (2005; 2007) for automatic pre-processing of large amounts of EModE text.
2. A few manual "corrections" were made to the C5 tagged output, either to match ProPOSEL's keys (as in *there's*_EX0+VBZ and *law's*_NN1+POS), or when an alternative syntactic analysis was preferred, for example (i) *for* as a subordinating conjunction rather than a preposition in: *for*_PRP *in*_PRP *that*_DT0 → ***for*_CJS** *in*_PRP *that*_DT0; and (ii) *off* as a particle rather than a preposition in: *shuffled*_VBN *off*_PRP → *shuffled*_VBN ***off*_AVP**. The finished version was saved as a text file.
3. This text file was then read in by the program and underwent the sequence of operations outlined in pseudocode in Listing 7.1: (i) the text was tokenized in the form of (word, tag) tuples; (ii) commas were tagged as minor boundaries `'|'` and a further set of punctuation marks `{ . : ; ? !}` were tagged as major boundaries `'||'`; (iii) the text was mapped to a tokenized First Folio edition of the same extract from NLTK's Project Gutenberg file; (iv) ProPOSEL was read in and transformed into a Python dictionary of compound keys and multiple values; (v) intersection between ProPOSEL's keys and the text object in the lookup process resulted in further prosodic-syntactic annotation of the text, including default content-function word tags; (vi) printouts of selected information in a user-friendly format were examined, *with surprising results*.

[12] Transforming *'tis* to *it's* does not affect syllable count and hence rhythm, unlike the transformation from *it's* to *it is* in Section 6.7.1 of this thesis

```
# hamlet_prosody2_rerun.py
# Compatible with NLTK 0.9.8
# 27/11/09

import nltk, re, pprint # main import statement for version
import copy
from nltk.tokenize import *
import itertools

tokenizer1 = LineTokenizer(blanklines='discard')
tokenizer2 = WhitespaceTokenizer()

# EModE version of text
textEModE = "To be , or not to be , that is the Question : Whether 'tis
Nobler in the minde to suffer The Slings and Arrowes of outragious Fortune
, Or to take Armes against a Sea of troubles , And by opposing end them :
to dye , to sleepe No more ; and by a sleepe , to say we end The Heart-ake
, and the thousand Naturall shockes That Flesh is heyre too ? 'Tis a
consummation Deuoutly to be wish'd . To dye to sleepe , To sleepe ,
perchance to Dreame ; I , there's the rub , For in that sleepe of death ,
what dreames may come , When we haue shuffel'd off this mortall coile ,
Must giue vs pawse . There's the respect That makes Calamity of so long
life : For who would beare the Whips and Scornes of time , The Oppressors
wrong , the poore mans Contumely , The pangs of dispriz'd Loue , the Lawes
delay , The insolence of Office , and the Spurnes That patient merit of the
vnworthy takes , When he himselfe might his Quietus make With a bare Bodkin
? Who would these Fardles beare To grunt and sweat vnder a weary life , But
that the dread of something after death , The vndiscouered Countrey , from
whose Borne No Traueller returnes , Puzels the will , And makes vs rather
beare those illes we haue , Then flye to others that we know not of . Thus
Conscience does make Cowards of vs all , And thus the Natiue hew of
Resolution Is sicklied o're , with the pale cast of Thought , And
enterprizes of great pith and moment , With this regard their Currants
turne away , And loose the name of Action ."

textEModE = textEModE.split() # tokenize the text

# PresE version of text with C5 PoS tag annotations
tagged = open ('C:\\...\\C5_hamlet_folio_punct_checked.txt', 'rU').read()

# Transformations on tagged version
tagged = tokenizer2.tokenize(tagged) # ['To_TO0', 'be_VBI', ',_,',..]
tagged = [tuple(index.split('_')) for index in tagged]
# [('To', 'TO0'), ('be', 'VBI'), (',', ','),..]
tagged = [list(index) for index in tagged]
# [['To', 'TO0'], ['be', 'VBI'], [',', ','],..]
for index in tagged: # Swap major & minor boundary markers for punctuation
if index[0] == ',':
     index.remove(index[1])
           index.append('|')
     elif index[0] in ['.', '?', '!', ':', ';']:
           index.remove(index[1])
           index.append('||')

final = list(zip(textEModE, tagged))



# Read in and transform ProPOSEL lexicon into Python dictionary
lexicon = open('C:\\...\\proPOSEL0408_final.txt', 'rU').read()
lexicon = [line.split('|') for line in list(tokenizer1.tokenize(lexicon))]
# Keys are immutable so tuples are used
lexKeys = [(index[0], index[1]) for index in lexicon]
# Nested lists: syll count; lex stress; CFP; DISC-stress mappings
lexValues = [[index[6], index[7], index[9], index[13]] for index in
lexicon] buildDict = dict(zip(lexKeys, lexValues))

# Performing the lookup
final2 = [((index[1][0].lower()), index[1][1]) for index in final] # Lower
```

```
case version required for ProPOSEL lookup
final4 = copy.deepcopy(final) # [...('To', ['To', 'TO0']), ('be', ['be',
'VBI'])...]

for x, y in itertools.izip(final2, final4):
     if x in buildDict.keys(): # if tuple format matches ditionary keys
           y[1].append(buildDict[x]) # append corresponding values to list
           format
     else:
           y[1].append('No match')

# Obtain printout
for line in final4:
     if line[0] in ['.', '?', '!', ':', ';', ',']:
            print line[0], line[1][1]
     elif line[1][2] == 'No match':
            print line[0], line[1][1]
     else:
            print line[0], line[1][1], line[1][2][2], line[1][2][3]
```

Listing 7.1: Program for automatic annotation of prosodic-syntactic features via ProPOSEL

7.3.2. True and false boundary predictions

There are 36 phrase break annotations corresponding to punctuation in the version of Hamlet's soliloquy used in this study. Any rule, therefore, which interprets punctuation as a phrase break feature, will have good recall: 100% in this case because the gold standard is based on punctuation alone. When the text is annotated with default content-function word tags from ProPOSEL, we find that 88.89% of these boundaries are *also* chink-chunk scenarios; but we also find that a chink-chunk rule dependent on these default settings over-predicts by inserting 50 extra boundaries, *false positives* of uncertain validity. CFP output (in horizontal format) for the final sentence looks like this (Example 7.1), with extra boundaries highlighted in **bold**.

Example 7.1

Thus_C Conscience_C does_C make_C Cowards_C **|** of_F vs_F all_F ,_| And_F thus_C **|** the_ F Natiue_C hew_C **|** of_F Resolution_C Is_C sicklied_No match o're_F ,_| with_F the_F pale_C cast_C **|** of_F Thought_C ,_| And_F enterprizes_C **|** of_F great_C pith_C **|** and_F moment_C ,_| With_F this_F regard_C **|** their_F Currants_C turne_C away_C ,_| and_F loose_C **|** the_F name_C **|** of_F Action_C ._||

Readers may decide which of these extra boundaries they might use themselves. It seems more natural to preserve syntactic cohesion in the following word clusters rather than chop them up into f-groups: *the native hue of resolution*; *the pale cast of thought*; *great pith and moment*; *the name of action*. On the other hand, falling rhythm makes *enterprises* and *regard* good candidates for a boundary of sorts: the in-line caesura. The word *resolution* is a similar case: it comes at the end of a line and its lexical stress pattern in ProPOSEL reveals two falls `|2010|`.

7.3.3. An accidental insight

A potential correlation was spotted, quite by accident, between words containing certain sounds and prosodic phrase boundaries in the outputs from Listing 7.1. The set of DISC phonetic transcriptions is peculiar in that it represents diphthongs and triphthongs (plus the long vowel in *or*) by the numerals 1 to 9; and these somehow stand out from the crowd. On inspection, it was noted that twelve out of thirty-three line ends in this extract contain diphthongs or triphthongs: {consummation; coile; life; time; delay; takes; make; beare; life; borne(?); moment; away}; and furthermore, that the same sounds co-occur with marked caesuras {dye; heart-ake; I; pawse; o'er} and with potential (unmarked) caesuras such as those already discussed: *nobler* and *minde* (7.1) plus *enterprizes* (7.2.1). Table 7.2. shows some example outputs.

Complex vowels in pre-boundary words	***Complex vowels at in-line caesuras***
: ||	, |

For CJS F 'f$R:1	And CJC F '{nd:1
who PNQ F 'hu:1	***enterprizes NN2 C 'En:1 t@:0 pr2:0 zIz:0***
would VM0 F 'wUd:1	of PRF F 'Qv:1
beare VVI C 'b8R:1	great AJ0 C 'gr1t:1
the AT0 F 'Di:1	pith NN1 C 'pIT:1
Whips NN2 C 'wIps:1	and CJC F '{nd:1
and CJC F '{nd:1	**moment NN1 C 'm5:1 m@nt:0**
Scornes NN2	, |
of PRF F 'Qv:1	With PRP F 'wID:1
time NN1 C 't2m:1	this DT0 F 'DIs:1
, |	regard NN1 C rI:0 'g#d:1
	their DPS F 'D8:1
	Currants NN2 C 'kV:1 r@nts:0
	turne VVB C 't3n:1
	away AV0 C @:0 'w1:1
	, |

Table 7.2: Co-occurrence of complex vowels and boundaries in *Hamlet* extract

7.4. Intuiting non-traditional phrase break features from verse

Automatic phrase break classifiers for text-to-speech synthesis systems currently rely on syntactic (*e.g.* part-of-speech) and text-based (*e.g.* punctuation) features for recapturing and emulating human parsing and phrasing strategies encapsulated by *gold standard* phrase break annotations in speech corpora used for training and testing such classifiers. In this case, the annotations correspond to listeners' perceptions of pauses in the speech stream as speaker prosody differentiates between syntactically coherent clusters of words: the *chunking* phenomenon (Abney, 1991).

Experimental work in the rest of this chapter is based on observation and intuition: the presence of diphthongs and triphthongs at phrase breaks or *rhythmic junctures* in poetry (7.2) suggests that new categorical *prosodic* features for boundary prediction may be derived from lexical items which incorporate this subset of English vowels – henceforth referred to as complex vowels for convenience – in their canonical phonetic transcriptions. The following examples (Examples 7.2 and 7.3) from English binary verse illustrate this association.

Example 7.2

Tyger! Tyger! burning bright
In the forests of the night,
What immortal hand or eye
Could frame thy fearful symmetry?

(The famous opening stanza of Blake's *The Tyger*, *circa* 1794)

Example 7.3

The dove descending breaks the air
With flame of incandescent terror
Of which the tongues declare
The one discharge from sin and error.
The only hope, or else despair
  Lies in the choice of pyre or pyre –
  To be redeemed from fire by fire.

(Eliot's Pentecostal invocation in part IV of *Little Gidding*, 1942)

The term rhythmic juncture is used here to denote in-line caesuras and line ends; and the lexical items of interest are words which immediately precede these boundaries and which bear complex vowels, often in the primary syllable, in their Present Day British English (PresE) canonical forms for both spelling and pronunciation. A vowel is said to be *complex* when vowel quality changes (from initial to target quality) within a single syllable (Maidment, 2009). Our subset includes words like *fire* and *power* where syllabification is dubious (one syllable or two?) and where transcriptions for standard English pronunciation vary between lexica (*cf.* 5.4.3 and 5.4.4). In plain text view, some of these junctures are not physically represented, either by punctuation or by line and verse endings: these are the unmarked, in-line caesuras. Nevertheless, the following pre-boundary tokens with vocalic glides are posited for Blake's stanza: {*Tyger; bright; night; eye; frame*}; and for Eliot's: {*air; flame; declare; hope; despair; choice; pyre; fire*}. In the case of recital, such choices (one might even say *classifications*) reflect '…speakers' perceptions about the divisibility of text…' (Sinclair and Mauranen, 2006: xvi); in silent reading, they reflect *projected* prosody (Fodor, 2002).

The present study undertakes an investigation to assess the degree of correlation between words bearing gliding vowels and marked boundaries in a classic Early Modern English (EModE) literary text: Book I of Milton's *Paradise Lost*. Software tools from version 0.9.8 of NLTK, the Natural Language ToolKit (Bird *et al.*, 2009) and Natural Language Processing (NLP) techniques are used to tokenize the text and then annotate it with 'projected prosody' from ProPOSEL, a prosody and part-of-speech English lexicon (Chapter 5). The principal dataset is drawn from Dartmouth College's eText of the 1674 edition of the poem (Luxon, 2010). The second dataset is a readily available, modern English version of Book 1: the 1992 eText from Project Gutenberg, also distributed in NLTK's corpora; although this does not entirely reflect original punctuation in the 1667 and 1674 editions, it is assumed to be a reliable phrasing variant[13].

This chapter discusses: the use of punctuation as a boundary marker in previous studies based on literary corpora (7.4); the tokenization and classification of each word in the samples as a break or non-break (7.5); the further annotation of each word token with its phonetic transcription via ProPOSEL, plus pertinent similarities and differences between EModE and PresE pronunciation (7.6); significance testing of the correlation between complex vowels and boundaries in both samples using the chi-squared statistic (7.7); and telling examples in Book 1 of Paradise Lost where unmarked conceptual boundaries (i.e. in-line caesuras) are signified by complex vowels (7.8).

7.5. Punctuation as a prosodic template

The symbolic representation of pauses via punctuation has been used in a number of exploratory studies of stylistic evolution in EModE blank verse. Pause patterns in Shakespeare's work, originally obtained from inclusive counts for punctuation at designated within-line positions for each play (Oras, 1960), have recently been subjected to formal statistical analysis (Jackson, 2002) and found to be good guides to chronology: plays of the same period, and in some instances, chronologically adjacent plays, reveal progressive experimentation with the

[13] Jackson (2002) observes that '…agents of transmission may prefer heavy or light punctuation, [but] tend not to diverge too markedly in where they place the stops…' (§2)

placement of stops (*i.e.* punctuation) within the line. A similar phenomenon, that of increasing divergence between metrical (the lines of verse) and grammatical units in the Shakespearian chronology, is discussed in a much earlier paper (Langworthy, 1931). Here, a quotient is obtained by dividing the number of parallel line types (*e.g.* where independent clauses are wholly contained *within* a line) by the number of divergent types in a given play and findings show that, whereas for very early plays the quotient is relatively high (40.00 and above), for later plays like *Hamlet* and *Macbeth* it is much lower (4.21 and 1.89 respectively) and for very late plays like *The Winter's Tale*, lower still (0.47). The following extract from Act I, Scene VII of *Macbeth* illustrates naturalistic prosodic-syntactic chunking both *within* and *between* lines, simulated via shifting placement of marked caesuras, and verse-sentence divergence facilitated by enjambement (Example 7.4).

Example 7.4

Macb. If it were done, when 'tis done, then 'twer well,
It were done quickly: If th' Assassination
Could trammell vp the Consequence, and catch
With his surcease, Successe: that but this blow…

Langworthy (1931) observes that the poet '…write[s] his sentence[s] almost as though he had forgotten all about the line, and yet fulfills the line requirements with the off-hand ease of a supreme master of metrics'.

Turning now to *Paradise Lost*, Banks (1927) sets out to identify the 'prosodical devices' by which Milton '…makes the rhythms of his units of thought independent of the single lines and of each other, thus achieving the effect of irregular paragraphs'. Again, punctuation in the form of terminal and medial stops {*periods; colons; question* and *exclamation marks*} is used to delineate verse paragraphs; but Banks' real interest is in classifying these joints in the verse in terms of trigrams consisting of a stop bordered by antecedent and posterior syllables which may or may not carry a beat. He identifies two prosodic patterns – the first of which is high-profile – which reinforce the midline break in Milton's verse through accent inversion, rather in the way of magnets: *like accents repel!* Examples of these are tabulated below (Table 7.3), with line references for Book I.

Juncture Type	Example	Line

A stop between two accents	'…for ever **dwells! Hail**, horrors…'	250
A *de-accented* stop	'…hast**ened: as** when bands…'	675

Table 7.3: Inverted accents in **bold** reinforce midline phrase breaks in Milton's verse

The present study also aims to explore *prosodical devices* associated with phrase breaks in *Paradise Lost*: namely, to test the intuition that diphthongs and triphthongs act as *vocalic precursors* of boundaries. It is assumed that punctuation in the principal dataset is sufficiently representative of the poet's phrasing and that *all* punctuation is significant. Such assumptions are supported by precedent; the terms *punctuation* and *pauses* have been used interchangeably in studies considered in this section; and inclusive counts for punctuation have incorporated: (i) major *and* minor boundary types; (ii) and *medial* as well as terminal stops. A further point is that punctuation is a primary feature used in language models for the machine learning of task of phrase break prediction: Ingulfsen *et al* (2005) even make the point that '…punctuation is used by writers to indicate rhythm and pausing'.

Experimentation (7.7) to determine whether the co-occurrence of complex vowels and pauses in Book I of *Paradise Lost* is statistically significant is based on a boundary count which includes *all* line-terminals in the count, irrespective of whether they are marked by punctuation or not. The mechanics and justification for this are covered in Section 7.5 and revisited in Section 7.8, where other types of conceptual boundary are also discussed.

7.6. Issues of tokenization and phrase break classification

The author has experimented with two different approaches to tokenization. Initially, for the Gutenberg sample, CorpusReader and Tokenizer Classes in NLTK 0.9.8 were used to simultaneously read in the unprocessed contents of this eText of *Paradise Lost* and to store these contents as a nested list of line tokens: the variable `milton` in the commented code snippet in Listing 7.2. The first line of the poem is then accessed via its list index, in this case `milton[2]`; and slice notation is used to assign the whole of Book I to a variable of the same name – `book1` – and to access and print out the first complete sentence: `milton[2:18]` by way of illustration. As an aside, punctuation in the output from Listing 7.2 accords well with the same excerpt as it appears in an original 1674 edition of the poem, viewable as a *.jpg* image on the internet (Geraghty, 2003).

```
import nltk, re
from nltk.tokenize import *#import all Tokenizer Classes from tokenize package
tokenizer = LineTokenizer(blanklines='discard')#initialize LINE tokenizer
milton = tokenizer.tokenize(nltk.corpus.gutenberg.raw('milton-
paradise.txt'))# Read in & tokenize lines in one step
book1 = milton[2:800] # start and end LINE indexes for Book I of the poem


>>> for line in milton[2:18]: print line # gives us the first sentence
Of Man's first disobedience, and the fruit
Of that forbidden tree whose mortal taste
Brought death into the World, and all our woe,
With loss of Eden, till one greater Man
Restore us, and regain the blissful seat,
Sing, Heavenly Muse, that, on the secret top
Of Oreb, or of Sinai, didst inspire
That shepherd who first taught the chosen seed
In the beginning how the heavens and earth
Rose out of Chaos: or, if Sion hill
Delight thee more, and Siloa's brook that flowed
Fast by the oracle of God, I thence
Invoke thy aid to my adventurous song,
That with no middle flight intends to soar
Above th' Aonian mount, while it pursues
Things unattempted yet in prose or rhyme.
```

Listing 7.2: NLTK's `LineTokenizer()` captures each line of verse in the Gutenberg eText as a separate token of type `string`

Listing 7.2 provides a solution for preserving verse *form* during tokenization. The next step is to transform `book1` so that every word in a line is captured as a separate token which can eventually be counted; each of these tokens is then classified as a break or a non-break, on the basis of two break indicators: associated punctuation and/or line terminal status. For the Gutenberg text, this was initially accomplished using NLTK's `WhitespaceTokenizer()`, which captures any attendant punctuation as part of each word token and thus facilitates the process of break classification. As an example, Listing 7.3 displays three phrase break tokens highlighted in bold: *Chaos*; *or* and *more*.

```
>>> for line in book1[8:11]: print line # Python lists start at 0
['In', 'the', 'beginning', 'how', 'the', 'heavens', 'and', 'earth']
['Rose', 'out', 'of', 'Chaos:', 'or,', 'if', 'Sion', 'hill']
['Delight', 'thee', 'more,', 'and', "Siloa's", 'brook', 'that', 'flowed']
```

Listing 7.3: NLTK's `WhitespaceTokenizer()` captures 3 break tokens in lines 9 to 11 of the Gutenberg eText

However, an alternative approach has since been used and has now been applied to both datasets in this study. The customised verse tokenizer in Listing 7.4 uses a regular expression (*cf.* Brierley and Atwell, 2009 for step-by-step decomposition and explanation of this regular expression) to differentiate word-internal from normal punctuation and effectively combats problems arising from house style punctuation, as in these pauses in lines 27-28 of the Gutenberg variant (Example 7.5).

Example 7.5

Say first--for Heaven hides nothing from thy view,
Nor the deep tract of Hell--say first what cause

Outputs from both the `WhitespaceTokenizer()` (labelled `test`) and the regular expression tokenizer (labelled `paradise`) are juxtaposed in Listing 7.4, where the existing data structure for `book1` undergoes further nesting to tokenize individual elements within each line.

```
import nltk, re
from nltk.tokenize import *
tokenizer = LineTokenizer(blanklines='discard')

# Read in & tokenize lines in one step
milton = tokenizer.tokenize(nltk.corpus.gutenberg.raw('milton-
paradise.txt'))

book1 = milton[2:800] # start and end LINE indexes for Book I of the poem
```

```
white = WhitespaceTokenizer()
test = [white.tokenize(index) for index in book1] # Tokenize on whitespace

# INSTANTIATE A CONTAINER AND APPLY A REGULAR EXPRESSION TOKENIZER TO CAPTURE WORD
TOKENS AND PUNCTUATION TOKENS, PRESERVING WORD-INTERNAL PUNCTUATION SUCH AS
HYPHENATED FORMS 'sea-monster'

paradise = [] # becomes a deeply nested array
for line in book1:
     paradise.append(re.findall(r"\w+(?:[-']\w+)*|[-.]+|\S\w*", line))


# OUTPUTS

>>> for line in test[26:28]: print line

['Say', 'first--for', 'Heaven', 'hides', 'nothing', 'from', 'thy',
'view,']
['Nor', 'the', 'deep', 'tract', 'of', 'Hell--say', 'first', 'what',
'cause']

>>> for line in paradise[26:28]: print line

['Say', 'first', '--', 'for', 'Heaven', 'hides', 'nothing', 'from',
'thy', 'view', ',']
['Nor', 'the', 'deep', 'tract', 'of', 'Hell', '--', 'say', 'first',
'what', 'cause']
```

Listing 7.4: Comparative outputs (**in bold**) from two different approaches to tokenization for the Gutenberg eText

As stated, the author has used the customised verse tokenizer for the count. Turning now to the principal dataset, Listing 7.5 operates on Dartmouth's eText and sorts all word tokens into different bags for breaks and non-breaks via a series of steps: (i) all 798 line terminal tokens are collected in `ends` and then subdivided on presence or absence of attendant punctuation (the containers `ends_punct` and `ends_nonpunct`); (ii) the container `minus_ends` is then created where line terminal word and punctuation tokens have been removed; (iii) a `for` loop captures medial breaks in `minus_ends` and then excludes them from consideration before the final iteration bags remaining tokens as non-breaks, ignoring punctuation tokens.

```
import nltk, re, copy
from nltk.tokenize import *
tokenizer = LineTokenizer(blanklines='discard')

# Dartmouth College version of Book 1 of Paradise Lost, 1674 edition
milton = open('...dartmouth_1674.txt', 'rU').read()#read in Book 1 as a string
book1 = tokenizer.tokenize(milton)

# INSTANTIATE A CONTAINER AND APPLY A REGULAR EXPRESSION TOKENIZER TO CAPTURE WORD
TOKENS AND PUNCTUATION TOKENS, PRESERVING WORD-INTERNAL PUNCTUATION SUCH AS
```

```
HYPHENATED FORMS 'sea-monster'
paradise = []
for line in book1:
     paradise.append(re.findall(r"\w+(?:[-']\w+)*|[-.]+|\S\w*", line))

# (i) CAPTURE THE LAST 2 TOKENS, WHICH COULD BE WORD + PUNCT OR ELSE 2 WORDS
ends = [index[-2:] for index in paradise]
ends_punct = [] # initialises container for end-stopped line-terminal word tokens
ends_nonpunct = [] # initialises container for run-on line terminal word tokens

for index, item in enumerate(ends):
    if '.' in item[-1]: # if the line terminates with punctuation...
        ends_punct.append((index, item[-2])) #...append previous word token
    elif ',' in item[-1]: ends_punct.append((index, item[-2]))
    elif ';' in item[-1]: ends_punct.append((index, item[-2]))
    elif ':' in item[-1]: ends_punct.append((index, item[-2]))
    elif '?' in item[-1]: ends_punct.append((index, item[-2]))
    elif '!' in item[-1]: ends_punct.append((index, item[-2]))
    elif ')' in item[-1]: ends_punct.append((index, item[-2]))
    elif '--' in item[-1]: ends_punct.append ((index, item[-2]))
    elif '"' in item[-1]: ends_punct.append((index, item[-2]))
    else: ends_nonpunct.append((index, item[-1])) # append terminal word token

# (ii) REMOVE LINE TERMINAL WORD AND PUNCTUATION TOKENS
minus_ends = []
for index, item in enumerate(paradise):
    if '.' in item[-1]: minus_ends.append((index, item[:-2]))
    elif ',' in item[-1]: minus_ends.append((index, item[:-2]))
    elif ';' in item[-1]: minus_ends.append((index, item[:-2]))
    elif ':' in item[-1]: minus_ends.append((index, item[:-2]))
    elif '?' in item[-1]: minus_ends.append((index, item[:-2]))
    elif '!' in item[-1]: minus_ends.append((index, item[:-2]))
    elif ')' in item[-1]: minus_ends.append((index, item[:-2]))
    elif '--' in item[-1]: minus_ends.append((index, item[:-2]))
    elif '"' in item[-1]: minus_ends.append((index, item[:-2]))
    else: minus_ends.append((index, item[:-1]))

# (iii) CAPTURE MEDIALS & BAG REMAINING WORD TOKENS AS NON-BREAKS
minus_ends2 = copy.deepcopy(minus_ends) # changes to copy won't affect original
medials = []# initialises container for word tokens marked as caesuras
non_breaks = []# initialises container for remaining non-break word tokens


for index, item in minus_ends2:
    for i, v in enumerate(item):
        if v in [',', '.', ')', '"', '!', '?', ':', ';', '--']:
            medials.append((i,  item[i  -  1]))  #  append  token  prior  to
punctuation
            del item[i - 1] # remove medial break token from line in minus_ends2

for index, item in minus_ends2:
    for i, v in enumerate(item):
        if v in [',', '.', ')', '"', '!', '?', ':', ';', '--', "'"]:
            pass # ignore punctuation tokens
        else: non_breaks.append((i, v))
```

Listing 7.5: Collecting and sorting all word tokens in Dartmouth College's eText of Book I of *Paradise Lost* into 5 different bags: (1) all line terminals; (2) end-stopped terminals; (3) run-on terminals; (4) marked caesuras; (5) non-breaks

The counts presented in Section 7.7 of this chapter are the true counts for (i) this particular version of the corpus and (ii) this particular solution for tokenizing blank verse. Even though we are ostensibly working with the same *poem* in the Dartmouth and Gutenberg eTexts, we are not working with the same *text* – or dataset – and subtle differences do emerge which affect the overall counts (but *not* the experimental outcome) for each version. One of the frequent culprits here is hyphenated forms: there are more of them in the Gutenberg version, hence reducing the word count for this dataset (*cf*. unshaded rows in Table 7.4). Another occasional difference is the representation of elisions – although it is unlikely that a modern reader familiar with the rhythms of blank verse would let such differences spoil the beat (*cf*. shaded rows in Table 7.4).

Gutenberg 311	And broken chariot-wheels. So thick bestrown,	6 word tokens
Gutenberg 340	Waved round the coast, up-called a pitchy cloud	8 word tokens
Gutenberg 460	In his own temple, on the grunsel-edge,	7 word tokens
Dartmouth 311	And broken Chariot Wheels, so thick bestrown	7 word tokens
Dartmouth 340	Wav'd round the Coast, up call'd a pitchy cloud	9 word tokens
Dartmouth 460	In his own Temple, on the grunsel edge,	8 word tokens
Gutenberg 223	"...In billows, leave **i' th'** midst a horrid vale..."	9 word tokens; 10 syllables intended
Dartmouth 223	"...In billows, leave **i'th'** midst a horrid Vale..."	8 word tokens; exactly 10 syllables

Table 7.4: Utterances which are *virtually* prosodically identical in the two datasets have different word counts

7.7. Projecting prosody onto text via ProPOSEL

ProPOSEL is a prosody and part-of-speech English lexicon of 104049 word forms, where each entry is mapped to a series of fields holding phonetic, syntactic and prosodic information about that word form. Fields of immediate interest to this study are (1) and (13): the headwords and DISC syllabified phonetic transcriptions which, unlike the more familiar International Phonetic Alphabet (IPA) and SAM-PA, use a single character to represent each phonological segment, irrespective of its complexity. Table 7.5 illustrates the distinctive symbolic equivalents for complex vowels in DISC which are so easy to spot.

Diphthong	SAMPA	DISC	Example	Example DISC Transcription
✓	/eI/	1	day	/ d1 /

✓	/aI/	2	night	/ n2t /
✓	/OI/	4	boy	/ b4 /
✓	/@U/	5	no	/ n5 /
✓	/aU/	6	now	/ n6 /
✓	/I@/	7	here	/ h7 /
✓	/e@/	8	there	/ D8 /
✓	/U@/	9	sure	/ S9 /

Table 7.5: Comparative representations of SAM-PA and DISC phonetic transcriptions for diphthongs in Received Pronunciation in English

ProPOSEL was originally designed for the target application of phrase break prediction, for compatibility with Python and NLTK, and for linkage with speech corpora. Projecting *a priori* linguistic knowledge from this lexicon onto corpus text is accomplished automatically. The Python programming language has a dictionary mapping object with entries in the form of (key, value) pairs, and this syntax is exploited by transforming ProPOSEL into a Python dictionary, with headwords (the *keys* in this case) mapped to an array of values from selected fields. During lookup, word tokens in the corpus acquire values associated with matching dictionary keys. In this way, the contents of each bag created in Listing 7.5 have now been tagged with prosodic annotations for further analysis. Table 7.6 shows the first six line terminal breaks in `ends_punct` after intersection with an instance of ProPOSEL holding the following symbolic values: syllable count (field 7); lexical stress pattern (field 8); content-function word tag (field 10); DISC transcription (field 13); stressed and unstressed values mapped to DISC syllable transcriptions (field 14).

[['woe', ['1', '1', 'C', "'w5", "'w5:1"]],
['seat', ['1', '1', 'C', "'sit", "'sit:1"]],
['seed', ['1', '1', 'C', "'sid", "'sid:1"]],
['song', ['1', '1', 'C', "'sQN", "'sQN:1"]],
[‘rhime’, 'rhyme', ['1', '1', 'C', "'r2m", "'r2m:1"]],
['pure', ['1', '1', 'C', "'pj9R", "'pj9R:1"]],..]

Table 7.6: Prosodic and syntactic annotations acquired via intersection with ProPOSEL for the first six end-stopped line terminals in Dartmouth College's eText of Book I of *Paradise Lost*, where tokens bearing complex vowels appear in **bold**.

7.7.1. What about the Great Vowel Shift?

In this study, present day pronunciation is projected onto EModE text – which is normally what contemporary readers/speakers do anyway with literary classics of this period – and findings are based on canonical forms: the eight diphthongs, plus the triphthongs, of Received Pronunciation (Roach, 2000: 21-24), or standard English speakers' '*wacky vowels*' (BBC, 2010). Some of these sound patterns would not have been used in Milton's day. The so-called Great Vowel Shift was a sound change over a prolonged period (roughly 1500 to 1800) that affected long vowels in English, such that their place of articulation shifted upwards, a process complicated by regional variation.

Barber (1997:139-40) offers a possible pronunciation for an extract from *To His Coy Mistress* by one of Milton's contemporaries, the poet Andrew Marvell. His transcriptions (presented in their equivalent SAM-PA forms in this section) for diphthongs in the following words: *time*; *coyness*; *down* accord with PresE pronunciations, while transcriptions for *day* and *no* simply indicate long vowels, the latter only becoming diphthongized in the late eighteenth century (ibid:107). An online simulation for EModE (Menzer, 2000) also suggests that the diphthong /aU/ in *loud* would have sounded much the same with an advanced speaker in 1650 as it does today but that the vowel in *name* was still in flux and reminiscent of the French sound *même*; this agrees with Barber's transcription for *day*: /dE:/. Another online simulation suggests that words like *time/bite* and *now/loud* did contain diphthongs but that these sounded more like hybrids of the combination *but* and *beet* and the combination *but* and *boot* respectively (Rogers, 2000).

Phonetic transcriptions in ProPOSEL show English vowels still in flux today and variation in source pronunciation lexica. The SAM-PA and CELEX transcriptions in fields 4 and 13 – derived from CUVPlus (Pedler and Mitton, 2002) and CELEX (Baayen *et al.*, 1996) – are in agreement for the following instance of the diphthong /U@/ in *pure*; interestingly, the CELEX notation for *pure* (*cf.* Table 6.6) incorporates a *y-glide* (*cf.* Bridges, 1921:24) and the same goes for the SAM-PA field: `/ pjU@R /`. On the other hand, the word form *moor* is realised with a diphthong in one source lexicon (CUVPlus) and a monophthong in the other (CELEX): `/ mU@R /` versus `/ 'm$R /`; the same speaker may also happily switch from one variant to the other. Finally, triphthongs are particularly unstable, as shown in the mismatches in syllabification in Table 7.7: CELEX does not appear to use any triphthongs and therefore *fire* and *power* are bi-syllabic; the CUVPlus transcription for *fire* may be interpreted as a triphthong, given the syllable count, but on the same basis, the transcription for *power* may not. This variance, even in canonical forms, is part of our language today.

	Word form	**Syllable count**	**SAM-PA**	**DISC**
CUVPlus	fire	1	'faI@R	
CELEX	fire	2		'f2-@R
CUVPlus	power	2	'paU@R	
CELEX	power	2		'p6-@R

Table 7.7: Instances of variant syllabification and phonetic transcription for the same orthographic form in pronunciation lexica show English vowels still in flux today

7.8. Significance testing: the correlation of complex vowels and phrase breaks

Sections 7.5 and 7.6 of this chapter have described how each word in Book I of *Paradise Lost* has been tokenized; then classified as a break or non-break, depending on the presence or absence of attendant punctuation, and as a further refinement, line-terminal status; and finally tagged with its modern-day phonetic transcription. Correspondence between the pronunciation of complex vowels in Milton’s day and ours has also been discussed (7.6.1).

Table 7.8 shows counts for the five different containers in Listing 7.5: {all line terminals; end-stopped terminals; run-on terminals; marked caesuras; non-breaks}, together with various counts for diphthongs and triphthongs obtained through dictionary lookup. These figures represent final counts after manual inspection and correction of totals for complex vowels due to unmatched items during lookup, where the latter generally comprise: proper nouns (*e.g. Nile; Sinai; Horonaim; Aonian*); and compounds (*e.g. sound-board; love-tale; straw-built; dove-like; night-*

founder'd), in addition to archaic words and forms (*e.g. compeer; scape; know'st; erewhile; extreams; battel; choyce*).

Queries	Containers	Counts
Number of LINE TERMINAL tokens	ends	798
Number of END-STOPPED lines	ends_punct	266
Number of RUN-ON lines	ends_nonpunct	532
Total number of MEDIAL BREAKS	medials	553
Number of NON-BREAKS which are *not* line-end tokens	non_breaks	4649
Total number of WORD TOKENS	ends + medials + non_breaks	6000
Total for TOKENS with attendant punctuation	ends_punct + medials	819
Total for TOKENS without attendant punctuation	ends_nonpunct + non_breaks	5181
Total number of BREAKS	ends + medials	1351
Total number of NON-BREAKS	non_breaks	4649
Total for unmatched diphthongs + triphthongs after ProPOSEL lookup	MANUAL INSPECTION OF: ends_punct; ends_nonpunct; medials; non_breaks	294
Total for unmatched diphthong_triphthong BREAKS after ProPOSEL lookup	MANUAL INSPECTION OF: ends_punct; ends_nonpunct; medials	106
Count for GLIDES as BREAKS, excluding unmatched items	ends_punct; ends_nonpunct; medials	419
Count for GLIDES as NON-BREAKS, excluding unmatched items	ends_nonpunct	874
Total count for GLIDES as BREAKS		419+106
Total count for GLIDES as NON-BREAKS		874+188
Total count for complex vowels		1587

Table 7.8: Shaded rows provide data for a chi-squared test based on a break count which includes *all* line terminals plus marked caesuras.

7.8.1. Applying the chi-squared test for collocation discovery

Based on figures from the shaded rows in Table 7.8 and entered in **bold** in Table 7.9, it is now possible to assign each word in the sample to one of four different categories and to compute and enter totals for each category in a 2 x 2 contingency table (*cf.* Table 7.9) ready for the chi-square test. The category label of diphthongs is used here to denote *all* complex vowels; and figures entered in bold

Table 7.9 juxtaposes observed and expected frequencies for all four categories obtained from the data in Table 7.8 and/or calculated from marginal totals in rows and columns for each category. Expected frequencies are given in *italics*; for example, the expected frequency for items in the sample which exhibit the following attribute-value pairings: diphthong {yes}; break {yes} is 357.34 (i.e. 1351 / 6000 * 1587).

GROUPS	**OUTCOMES**		**TOTALS**
	Breaks	**Non-breaks**	
Diphthongs	**525** *357.34*	**1062** *1229.66*	**1587**
No diphthongs	826 *993.66*	3587 *3419.34*	4413
TOTALS	**1351**	4649	**6000**

Table 7.9: Observed and expected frequencies are computed from the **raw counts** obtained in Listing 4.

We assume that the distributions resulting from observed (f_o) and expected frequencies (f_e) in the shaded area in Table 7.9 will be very similar: this is the null hypothesis $H_{o.}$ Then, if the value of chi-squared χ^2 according to the following formula exceeds some critical value, we can reject H_o and surmise that the observed distribution is unlikely to have occurred by chance, and that diphthongs and boundaries are not independent of each other.

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

In this case, the association between groups and outcomes is deemed to be highly significant: chi squared equals 138, with 1 degrees of freedom, and a two-tailed p-value or odds ratio which is less than 0.0001.

The break count in Table 7.8 for Dartmouth's eText of the 1674 edition of Book 1 is lower than that of the more heavily punctuated Gutenberg version: 1351 to 1447 respectively. Nevertheless, the Gutenberg text is a reliable phrasing variant, encapsulating an alternative parsing and phrasing strategy for the reader or speaker. It is of consequence, therefore, that the statistically significant correlation between complex vowels and phrase breaks is corroborated by this dataset. Experimental replication returns a chi-squared statistic of 123, with 1 degrees of freedom, and a two-tailed p-value of 0.0001.

7.9. Textual analysis of unmarked conceptual boundaries in Book 1 of Paradise Lost

This investigation is based on rhythmic junctures in Book I of *Paradise Lost* marked by punctuation and by line ends. That the latter represent conceptual boundaries and reflect *performance structure* (*cf.* Gee and Grosjean, 1983; Abney, 1992) in theatre or for recital is apparent in this directive on verse-speaking from the Royal Shakespeare Company (Hall, 2004: 28): '…[t]he end of each line is in fact a punctuation often more crucial than the regular punctuation itself'. An alternative view or segmentation of the text is implemented via XML markup in Durusau and O'Donnell (2002); their *sentence view* differs from traditional presentation, which preserves the integrity of each line (*cf.* the tokenization process in Section 7.5), and instead segments on punctuation, so that chunks often run from one line to the next and sometimes incorporate constituents from more than two lines. In the following example (Example 7.6) of verse-sentence divergence from lines 10-12 of Book I, segments (i.e. strings between <seg></seg> XML tags) reflect punctuation in the Raben[14] version.

Example 7.6

```
…or, </seg><seg> if Sion hill
Delight thee more, </seg><seg> and Siloa's brook that flowed
Fast by the oracle of God, </seg><seg> I thence
Invoke thy aid to my adventurous song, </seg><seg>
```

Run-on lines are common in blank verse. Borrowing terminology from Durusau and O'Donnell (*ibid.*), line terminals which are not end-stopped are members of *overlapping hierarchies*. They represent the logical relation of *intersection* between two different sets within the sentence: the metrical line and the prosodic-syntactic chunk. The token *down*, for example, in '…With hideous ruin and combustion down…' (*Paradise Lost*, Book I, line 46) is unmarked with punctuation in the original 1667 and 1674 editions of the poem, as well as the Project Gutenberg eText, and exhibits this kind of duality. It is also part of a wider

[14] The eText used in Project Gutenberg's Paradise Lost was originally created by Dr. Joseph Raben of Queen's College, NY circa 1964-5.

context: the sentence container spanning lines 44 – 49 (*cf.* Table 7.10), where the majority of terminals are run-on and carry diphthongs or triphthongs.

1667 and 1674 editions	**Project Gutenberg eText**
…Him the Almighty Power Hurld headlong flaming from th' Ethereal Skie With hideous ruine and combustion down To bottomless perdition, there to dwell In Adamantine Chains and penal Fire, Who durst defie th' Omnipotent to Arms.	'…Him the Almighty Power Hurled headlong flaming from th' ethereal sky, With hideous ruin and combustion, down To bottomless perdition, there to dwell In adamantine chains and penal fire, Who durst defy th' Omnipotent to arms…'

Table 7.10: Phrasing in 17th. century editions of *Paradise Lost* is more open-ended

While Raben's version faithfully reflects poetic elisions (*th'etheral*; *th'Omnipotent*), it is more prescriptive in its punctuation such that, in sentence view, *down* would be assigned to a different segment from the 17th century versions. In the latter, the token *down* is highly ambiguous; syntactically, it is probably part of the compound preposition *down to* and attached to the subsequent noun phrase *bottomless perdition*, but the absence of punctuation seems to preserve an almost uncapturable, long-distance syntactic and semantic relationship to the verb *Hurld*, in which case, *down* would be a particle as in: *Satan was hurled down from heaven*. By twice separating *down* from *hurled*, with commas after *sky* and *combustion*, Raben has edited out some poetic effects: *down* as a particle lost in space, as a long sound lamenting the terrible violence of Satan's severance from God.

The Fall – and Milton's depiction of it – is indelible from our imaginations; a recent stunning re-enactment is the opening sequence of Peter Jackson's film adaptation of *The Two Towers* (2002) and that long shot of the Balrog falling flaming from the bridge of Khazad-Dum into the pit of Moria. Images of falling abound in Book I. There is the famous Mulciber passage where again, a gathering of complex vowels and long vowels in the hinterland between lines delays the verse movement as we witness the protagonist's fall from grace – a beautiful slow-motion arc (Example 7.7).

> '…**thrown** by angry **Jove**
> **Sheer o'er** the crystal battlements: from **morn**
> To **noon** he fell, from **noon** to **dewy eve**,

A summer’s **day**…’

7.9.1. Caesuras as conceptual boundaries

The boundary concept in verse may be extended to caesuras or rhythmic junctures within the line, though it has not been possible to include all candidates in the boundary count for the present study because the location of *unmarked* caesuras is open to interpretation and we have no agreed gold standard to work from. Nevertheless, complex vowels may signal optimal phrase break opportunities within the line, especially when enjambement encourages the reader or speaker to process phrases like ‘…I thence / Invoke thy aid to my adventurous song…’ as one chunk (*cf.* XML segmentation in Example 7.6). Would chunking or pausing somewhere within this phrase enhance a reader’s or listener’s understanding? If so, where *is* the best place to pause? Is it after *thence* or is it after the diphthong-bearing *aid*?

One final extract (*cf.* lines 17-26 in Table 7.11) from Book I of *Paradise Lost* may serve to highlight how complex vowels signify conceptual boundaries which are pivotal to the parsing strategy for that sentence; and how the correlation of complex vowels and boundaries seems, in fact, to fit Saussure’s model of the sign: ‘…[a] linguistic sign is not a link between a thing and a name, but between a concept [signified] and a sound pattern [signifier]…’ (Saussure in Chandler, 2002:18). Diphthongs act as *precursors* or signifiers of phrase breaks.

1674 version	**1992 version**
And chiefly Thou O Spirit, that dost prefer	And chiefly thou, O Spirit, that dost prefer
Before all Temples th' upright heart and pure,	Before all temples th' upright heart and pure,
Instruct me, for Thou know'st; Thou from the first	Instruct me, for thou know'st; thou from the first
Wast present, and with mighty wings outspread	Wast present, and, with mighty wings outspread,
Dove-like satst brooding on the vast Abyss	Dove-like sat'st brooding on the vast Abyss,
And mad'st it pregnant: What in me is dark	And mad'st it pregnant: what in me is dark
Illumine, what is low raise and support;	Illumine, what is low raise and support;
That to the highth of this great Argument	That, to the height of this great argument,
I may assert Eternal Providence,	I may assert Eternal Providence,
And justifie the wayes of God to men.	And justify the ways of God to men.

Table 7.11: Again, phrasing in the most popular 17th. century edition of *Paradise Lost* is less directive than a contemporary edition

Punctuation is again more subtle in the 17th century version and assumes a poetic sensibility and a poetic ear. In the original, for example, *Dove-like* belongs both to *outspread* and to *sat'st*, whereas the modern edition eliminates one of these paths. Moreover, a comma after *Abyss* at the end of line 21 is perhaps redundant because we cannot produce a succession of sibilants {sat'st; vast; Abyss; mad'st} without slowing down. The section of interest, however, is lines 22-23, where both versions agree.

Assuming that punctuation represents the poet's phrasing, we are meant to pause at the comma in: '…What in me is dark / **Illumine,** what is low raise and support…' Nevertheless, despite the status of *dark* as a run-on line terminal, and despite its proximity to the marked boundary in *Illumine*, the syntax requires a break at this point; the bigram <adjective><verb> is unusual and the line-break alerts us to this fact. In the subsequent clause, we have a repetition of this uncommon template but instead of a line-break, we have two consecutive diphthongs: **low raise** inhibiting normal phonotactics. A *gold standard* phrasing of this section is hypothesized as follows: '…What in me is dark | Illumine, | what is low | raise and support; |…' Thus adjacency of complex vowels has been interpreted as a textual cue or *text-based feature* in a difficult syntactic context and in the absence of explicit permission to pause.

7.10. Concluding Comments

This study uses punctuation, as in previous work on pause patterns in English verse, plus line endings, as equivalents for gold standard phrase break annotations and discovers a significant correlation between complex vowels (i.e. diphthongs and triphthongs) and prosodic-syntactic boundaries, a result which is replicated in *two* naturalistic phrasing variants of the same poem. This finding is believed to have several implications. First, complex vowels (like punctuation itself) constitute a domain-independent phrase break feature. Thus, what works for verse may also work for prose; and the author will shortly report on similar findings in a parallel experiment for PresE using an extract from the Aix-MARSEC dataset (Chapter 8). Second, while punctuation is a top-performing phrase break feature, it does not capture all perceived prosodic-syntactic boundaries. The use of additional run-on line endings as conceptual boundaries substantiates these findings and also highlights the ambiguous status of some phrase break tokens as constituents of more than one syntactic grouping, where the groups are not always immediately adjacent, even in plain text view. The chapter also considers complex vowels as boundary precursors, as textual cues signifying optimal parsing and phrasing strategies, and enhancing understanding, for readers and speakers alike. Finally, the *prosodical devices* used deliberately or subconsciously by poets (Milton did say his verse was *unpremeditated*[15]) may provide generic insights into prosodic-syntactic chunking. Banks (6.4) detects accented and deaccented stops which can be parameterised for experiments with PresE speech corpora (*cf.* Aix-MARSEC); and he leaves us with an intriguing observation: that *units of thought* are *rhythmical*.

[15] *Paradise Lost*, Book 9, line 24

Chapter 8
Experimentation with ProPOSEL (Part 2): Significance Testing of Non-traditional *Prosodic* Phrase Break Features in a Corpus of Transcribed Speech from the Twentieth Century

8.1. Recapitulation

The previous chapter reported on a significant correlation between lexical items containing complex vowels and prosodic-syntactic boundaries in seventeenth century verse. Real-world knowledge of PresE canonical pronunciation from ProPOSEL was projected onto each word token in two different versions, constituting two phrasing variants, of *Paradise Lost*, Book 1. The chi-squared test for independence returned a two-tailed p-value of less than 0.0001 for the association of this vowel subset and phrase breaks in both samples. This led to speculation that Milton’s *unpremeditated* use of complex vowels – which slow down verse movement in *Paradise Lost* and thus generate rhythmic junctures – may represent a phrasing device habitual not just to poets but to native English speakers in general. In this chapter, concurrent work on a corpus of present-day British English speech corroborates these findings.

8.2. Research questions

Complex vowels may constitute a new predictive feature in phrasing models for English, especially since this feature is robust enough to tolerate ‘noisy’ variant phrasing strategies for the same text or speech (*cf.* 4.3.6): the statistically significant correlation between words carrying complex vowels and phrase breaks is corroborated by both datasets in Chapter 7.

The intuition that the presence of complex vowels in content words increases the likelihood of their being classified as breaks *comes* from poetry, where diphthongs and triphthongs seem to be associated with rhythmic junctures, and has been *tested* on poetry. Experimental work in this chapter sets out to determine whether this association holds good for: (i) ordinary (if formal) comtemporary British English speech; (ii) spontaneous as well as read speech (*i.e.* different genres); (iii) multiple speakers. In all cases, the datasets used merge information from the Spoken English Corpus and the Aix-MARSEC corpus project.

8.3. Hypothesising non-traditional phrase break correlates

A recent study by Ananthakrishnan and Narayanan (*cf.* 3.7) attempts to integrate the prediction of accents and boundaries based on combined feature streams (acoustic, lexical and syntactic) and finds that lexical syllable *tokens*, augmented with canonical stress labels derived from an open source pronunciation lexicon, are effective for accent detection but not for boundary prediction.

Ananthakrishnan and Narayanan conclude that syllable tokens are poorer indicators of boundary events than PoS tags. However, this conclusion is based *only* on word-final syllable tokens *minus* stress weightings for the phrase break prediction task; word-initial and medial syllables are automatically classed as non-breaks because they are never immediately followed by boundary tokens.

This thesis questions the assumption that non word-final syllabic nuclei (e.g. the second syllable in `seCURity`) have no influence on boundary placement and tests the hypothesis that complex vowels – i.e. diphthongs and triphthongs – might emerge as useful predictive features for phrase break models, irrespective of where they occur within a word. There is consensus within the ASR research community that pauses affect vowel durations in preceding words (Vergyri *et al.*, 2003). This thesis reverses the perspective on prepausal lengthening and asks to what extent a domain-independent feature like complex vowels may be said to *induce* boundaries.

The prosody and PoS English lexicon (*cf.* Chapter 5) addresses the perceived need (*cf.* 5.1) for prosodic features to complement syntax and punctuation in phrase break models, thus extending the knowledge source for this classification task. Furui (2009) has stated that improvements in ASR *depend* on better knowledge sources; and there is a current trend in various application domains towards supplementing raw training data with *a priori* knowledge, where hitherto little real-world knowledge has been assumed on the part of the learning mechanism (*cf.* PASCAL-2 (2008); CFP for IJCAI 2009 on User-Contributed Knowledge and Artificial Intelligence). The survey in Chapter 2 diagnoses a deficiency of *a priori* linguistic knowledge of prosody in feature sets typically used for automatic phrase break prediction, whereas competent human readers will habitually project prosody onto text and treat this as part of the input. This thesis contends that human readers may use the sound patterns inherent in complex vowels as *linguistic signs* for phrase breaks in as yet undefined contexts. It also contends that such signs are domain-independent, that they can be extracted from the lexicon and, just like PoS tags, can be projected onto *any* corpus and can subsequently be presented as input features to the phrase break classifier.

8.4. Automatic annotation of composite SEC and Aix-MARSEC Section C dataset

To investigate the correlation between complex vowels and phrase breaks in contemporary British English speech, an extract from the Aix-MARSEC corpus was automatically tagged with shallow parse features and canonical phonetic transcriptions from ProPOSEL. A chi-squared test was then used to determine whether this correlation is statistically significant or not. The experimental dataset was the same as that used in previous studies (*cf.* 3.3): a BBC radio recording from the 1980s of a Reith lecture in Section C of the corpus. Approximately half of this was sampled, with original phrase break annotations from Gerry Knowles, plus a short section inter-annotated by Knowles and Bryony Williams. Illustrative examples are also taken from a previously used dataset: informal news commentaries in sections A08 and A09 of the corpus.

Preparing the dataset prior to dictionary lookup was non-trivial and involved several stages. The first task was to map annotation tiers in overlapping subfiles in the Aix-MARSEC sample in order to label each word as a break or non-break (8.4.1). Word and phrase break classifications in Aix-MARSEC were then merged with corresponding PoS-tagged text in the Spoken English Corpus, where discrepancies intervene: compounds and abbreviations are handled differently in both datasets, for example (8.4.2). Next, the corpus was re-tagged with the PoS tag scheme used in the lexicon (8.4.3) i.e. a discriminating tagset (LOB) was collapsed into a sparser one (C5). Finally, desired information from the lexicon was projected onto the dataset by matching up word-C5 pairings (8.4.4).

8.4.1. Mapping tiers in Aix-MARSEC

The Aix-MARSEC Corpus has multi-level prosodic annotation tiers aligned with the speech signal; the two tiers used in this study are for plain text plus intonation units (IUs) delineated by phrase break mark-up / **|** /. The SAMP-PA transcriptions from the syllables tier were not used in this study because the focus is on predictive features derived from speaker-independent and domain-independent *citation* forms in ProPOSEL which can be superimposed on *any* unseen English text – for example, seventeenth century English verse (*cf*. Chapter 7).

Each section in Aix-MARSEC is split up into a series of much smaller, overlapping TextGrid files. Merging the text and IUs tiers was therefore accomplished on a file-by-file basis, using interval tokens to retrieve a match between tiers. The resulting list objects were concatenated in a final list – `listAllText` – ready for merger with the corresponding file in the Spoken English Corpus (SEC) to capture PoS-tags.

8.4.2. Merging Aix-MARSEC and SEC files

The target data structure for dictionary lookup (8.4.4) is a nested list where each index holds values for: word token; break class; punctuation; and PoS-tag. Capturing PoS tags from SEC entailed looping over two parallel lists of unequal length – `listAllText` and a list of `word_PoS` pairings from SEC – a process complicated by the fact that compound words are represented differently in both datasets, and furthermore, that punctuation in SEC does not always correspond to boundaries or placeholders in Aix-MARSEC. Such problems are exemplified in Table 8.1 (from section A09 of the corpus), where we find different representations for the compound adjective: *cross-ethnic*; variant phrasing for the fragment: *who two years ago*; no apparent placeholder in Aix-MARSEC following the boundary after *ago*; and no punctuation in SEC after the word *together*, which is marked as a phrase break in Aix-MARSEC.

Aix-MARSEC	**SEC**	
['ethnic', '48.69', '|']	JJ	ethnic
['#', '48.74', 'P']	,	,
['cross', '49.12', 'non-break']	**JJ**	**cross-ethnic**
['ethnic', '49.53', '|']	,	,
['#', '49.62', 'P']	CC	and
['and', '49.88', 'non-break']	JJ	political
['political', '50.41', 'non-	,	,

<table>
<tr><td>break']</td><td>NNS</td><td>parties</td></tr>
<tr><td>['parties', '50.88', '|']</td><td><b>WP</b></td><td><b>who</b></td></tr>
<tr><td>['#', '51.39', 'P']</td><td>,</td><td>,</td></tr>
<tr><td><b>['who', '51.59', 'non-break']</b></td><td><b>CD</b></td><td><b>two</b></td></tr>
<tr><td><b>['two', '51.73', 'non-break']</b></td><td><b>NNS</b></td><td><b>years</b></td></tr>
<tr><td><b>['years', '52.04', 'non-break']</b></td><td><b>RB</b></td><td><b>ago</b></td></tr>
<tr><td><b>['ago', '52.44', '|']</b></td><td>,</td><td>,</td></tr>
<tr><td>['came', '52.70', 'non-break']</td><td>VBD</td><td>came</td></tr>
<tr><td><b>['together', '53.12', '|']</b></td><td><b>RB</b></td><td><b>together</b></td></tr>
<tr><td><b>['#', '53.17', 'P']</b></td><td>TO</td><td>to</td></tr>
<tr><td>['to', '53.34', 'non-break']</td><td></td><td></td></tr>
</table>

Table 8.1: Transcriptions of the same utterance in two different versions of the corpus exhibit variant phrasing.

8.4.3. Mapping between PoS tag sets using ProPOSEL

List indices in the object `listAllText` have now acquired PoS tags and, if present, punctuation from the semi-automatic process just described. However, the recommended lookup strategy with the prosody and PoS lexicon is via compound dictionary keys comprising `word_C5` pairings. A range of tagsets (Penn, LOB and C7) were mapped to C5 as part of lexicon build; and ProPOSEL's software tools provide solutions for mapping between schemes (Chapter 5). In the present study, a more discriminating tagset, LOB, is collapsed into a sparser scheme: C5. As part of this process, enclitics in LOB are re-formatted in a style compatible with the lexicon; instances such as: `['BEDZ', 'was', '>', 'XNOT', "n't", '<']` and `['WP', 'who', '>', 'HV', "'ve", '<']` are transformed into: `['BEDZ+XNOT', "wasn't"]` and `['WP+HV', "who've"]`.

8.4.4. Dictionary lookup and text annotation

Nested arrays in `listAllText` are finally augmented with domain knowledge of prosody (*i.e.* DISC fields in ProPOSEL) and coarse-grained syntactic information (default content-function word tags) via intersection with ProPOSEL. Listing 8.1 first builds an instance of the dictionary object `proPOSEL` with compound keys `word_C5` tuples mapped to selected values. Python's `itertools()` module is then used to loop through two parallel iterables: `listAllText` and `match`, a sequence of `word_C5` tuples from the same dataset. Items in the latter are compared against

ProPOSEL's keys; a successful match appends dictionary values associated with those keys to the parallel nested position in `listAllText`.

```
proPOSEL = dict(zip(lex_keys, lex_values))
match = [(index[0], index[5]) for index in listAllText]
for x, y in itertools.izip(match, listAllText):
    if x in proPOSEL.keys():
        y.append(buildDict[x])
    else:
        y.append('No match')
[tuple(line) for line in listAllText] # the final set of annotations
```

Listing 8.1: Intersection between the dictionary object `proPOSEL` and the sequence object `match` appends dictionary values to the parallel position in `listAllText`.

Inner lists in `listAllText` have now been augmented with content/function-word tags, DISC phonetic transcriptions and canonical stress weightings aligned with syllables (e.g. the lexical stress pattern `2010` assigned to the DISC transcription for the word *contribution*: `"kQn:`**`2`** `trI:`**`0`** `\'bju:`**`1`** `SH:`**`0`**).

8.5. Significance testing for Section C dataset

Each word in the sample was assigned to one of four different categories and counts for each category were entered in a 2 x 2 contingency table (Table 8.2) ready for the chi-square test. The category label of `diphthongs` is used here to denote *all* complex vowels. The total word count is simply the length `of listAllText` minus the count for unmatched items; these were not included in the final calculation and figures used in Table 8.2 reflect this.

GROUPS	OUTCOMES		
	Breaks	**Non-breaks**	
Diphthongs	**201**	298	**499**
No diphthongs	437	1357	1794
	638 (696 - 58)	1655	**2293** (2468 - 175)

Table 8.2: A 2 x 2 contingency table records the observed frequency distribution for target groups and outcomes from the corpus sample.

The chi-square test in this experiment determines whether the distribution resulting from observed frequencies in the shaded area in Table 8.2 is significantly different from the chance distribution anticipated from expected frequencies. The latter are calculated via marginal totals for rows and columns in the table: for example, the expected frequency for diphthongs classified as breaks is given by (638 / 2293) * 499. Table 8.3 presents observed (given in **bold**) versus expected frequencies (given in *italics* and expressed as whole numbers for clarity of presentation) for all four categories.

GROUPS	**OUTCOMES**	
	Breaks	**Non-breaks**
Diphthongs	**201** *139*	**298** *360*
No diphthongs	**437** *499*	**1357** *1295*

Table 8.3: Observed and expected frequencies are used to find the value of χ^2 in this test for independence.

These figures are then used to find the value of χ^2 according to the formula previously given (see 7.8.1.).

The null hypothesis H_o assumes that the distributions will be the same or that the difference will not exceed some critical value. In this case, however, H_o can be rejected because the association between groups and outcomes turns out to be extremely statistically significant: chi squared equals 49.28, with one degree of freedom, and a two -tailed p-value which is less than 0.0001. This p-value represents the odds ratio for achieving the same result through random sampling. Finally, since there are only *four* diphthong-bearing function words which are also classified as breaks in this sample (§4.2.3), we can hypothesize that the significant correlation is actually between diphthong-bearing *content* words and phrase breaks.

8.6. Etymology in Section C dataset

A further research question tested via the Section C dataset was whether or not there is any association between etymology and phrasing. For this experiment, every word in the sample was assigned to one of two groups: Old English or Latinate/Other. Classifications were made with reference to the Collins English Dictionary (Sinclair, 1994), where words derived from Old English, Norse, Frisian, Saxon *et cetera* were subsumed into the Old English group. Table 8.4 records the counts used in the significance test for this feature. Etymology was found to be highly correlated with phrasing, returning a chi-squared statistic of 456, with 1 degrees of freedom, and a two-tailed p-value of less than 0.0001 for the Section C data. One might hypothesise that words in the Latinate/Other category are more likely to be content words and to have richer prosodic attributes (*e.g.* a rhythmic profile that guarantees a beat) – hence their association with boundaries.

GROUPS	**OUTCOMES**		
	Breaks	**Non-breaks**	
Old English	**169**	1240	**1409**
Latinate	469	415	884
	638 (696 – 58)	1655	**2293** (2468 – 175)

Table 8.4: 2 x 2 contingency table for distribution of Old English versus Latinate/Other words in relation to phrase break annotations in the corpus

8.7. Significance testing on a multi-speaker dataset of spontaneous speech

So far, we have gathered empirical evidence from seventeenth century verse and read speech from the twentieth century which highlights a statistically significant correlation between words carrying complex vowels and phrase breaks in English via the chi-squared test for independence. This investigation is now extended to spontaneous speech, while reminding readers that the gold-standard phrase break annotations used still denote *intentional* as opposed to disfluent pauses.

8.7.1. Custom-built dataset

The dataset used to test the correlation between complex vowels and phrase breaks in the genre of spontaneous as opposed to read speech, and for multiple speakers instead of a single speaker, was custom-built to align word tokens and phrase break information from Aix-MARSEC, with syntactic information (*i.e.* LOB PoS-tags) from SEC and ProPOSEL (*i.e.* C5 PoS-tags), plus punctuation from SEC, plus shallow parse features (*i.e.* content-function word tags) and canonical phonetic transcriptions, again from ProPOSEL. The dataset of 7762 *word* tokens was compiled from informal news commentary in Section A of the corpus: it includes ten different speakers, both male and female, and two different annotators: Gerry Knowles and Briony Williams, and is outlined in Table 8.5. The algorithm used for this most recent dataset build is outlined in Section 8.9.

Section A file no.	**Word count**	**Break count**	**Speaker gender**	**Annotator**
A01	791	135	Female	Williams
A03	635	120	Male	Williams
A04	984	283	Male	Knowles
A05	803	200	Male	Knowles
A06	827	126	Male	Williams
A07	714	163	Male	Knowles
A08	629	120	Male	Williams
A09	789	199	Male	Knowles
A10	801	132	Male	Williams
A11	789	147	Male	Knowles

Table 8.5: Overview of dataset used.

8.7.2. Obtaining the counts

Word and phrase break totals for each Section A sub-file in Table 8.5 constitute initial values for a 2 x 2 contingency table exploring the relationship between two distinct *groups*: diphthong-bearing words versus words with no diphthong (where the label 'diphthong' stands for *all* complex vowels); and two distinct *outcomes*: breaks versus non-breaks. Word counts were obtained by subtracting the break count (number of pauses) from the length of each file. Each word token was then classified as a break or non-break, depending on whether or not it was followed by a pause.

The total counts for diphthong and non-diphthong-bearing words were generated automatically for the most part but subject to manual inspection where prosodic information from ProPOSEL was (or appeared to be) missing. Missing information was due to a variety of factors. The dataset is spattered with proper nouns which do not appear in the lexicon. Furthermore, there are omissions passed down from source lexica: the noun *hijackings* from A08 does not appear as a plural in ProPOSEL, for example; and while the verb *rely* (in A11) carries a lexical stress pattern generated from one source, it has no values for fields 13-15 simply because they are generated from an alternative source which, surprisingly, does not include that word. Finally, there are some ‘freaks of nature’ such as the misspelling of *disillusioned* in Section A09 of the corpus: (A09|**dissillusioned**|non_break|AJ0|No_match). There are, in fact, several opportunities for a match here in ProPOSEL, depending on whether the word has been tagged in context as an adjective, past participle or past preterite.

8.7.3. Running the chi-squared test

Four counts were used to populate each 2 x 2 contingency table: word and break counts from Table 8.5 and total counts for diphthong-bearing (content and function) word *breaks* versus diphthong-bearing (content and function) word *non-breaks*. The remaining counts were generated from these as in this example (Table 8.6) from Section A09.

GROUPS	**OUTCOMES**		
	Breaks	Non-breaks	Totals
Diphthongs	**57**	**129**	186
No diphthongs	142	461	603
Totals	**199**	590	**789**

Table 8.6: A 2 x 2 contingency table records the observed frequency distribution for target groups and outcomes from corpus sample A09.

The chi-square test in this experiment determines whether the distribution resulting from observed frequencies in the shaded area in Table 8.6 is significantly different from the chance distribution anticipated from expected frequencies. The latter are calculated via marginal totals for rows and columns in the table: for example, the expected frequency for diphthongs classified as breaks is given by (199 / 789) * 186.

8.8. Discussion of results for multi-speaker corpus of spontaneous speech

Table 8.7 presents a summary of the findings. On the evidence of this study, the correlation between words carrying complex vowels and phrase breaks in English is a very significant stylistic feature of some speakers (at least 50%) but not others.

Section A	**Ratio: words**	**Value of χ^2**	**2-tailed**	**Significant?**

file number	**to breaks**		**p-value**	
A01	5.86 : 1	0.356	0.5510	No
A03	5.29 : 1	0.095	0.7585	No
A04	3.48 : 1	25.354	< 0.0001	Yes
A05	4.02 : 1	15.976	< 0.0001	Yes
A06	6.56 : 1	1.358	0.2439	No
A07	4.38 : 1	10.947	0.0009	Yes
A08	5.24 : 1	30.090	< 0.0001	Yes
A09	3.97 : 1	3.795	0.0514	Not quite
A10	6.07 : 1	0.873	0.3502	No
A11	5.37 : 1	7.885	0.0050	Yes

Table 8.7: Results per file for the chi-squared test.

The presence or absence of this habit of speech seems to be independent of speaker gender and discernible (albeit subconsciously) to different listeners: both Knowles' and Williams' phrase break annotations are consistent with the findings. There also seems to be a link to phrasing density: on balance, the significant correlation occurs with speakers who pause more often. The densest phrasing occurs in A04, where dramatic reportage covers war-torn El Salvador. What is interesting in these findings is: (i) there is a stark contrast between these two types of speaker; and (ii) a multi-speaker corpus of spontaneous speech corroborates findings from previous experiments (8.5), where the datasets might be described as 'composed speech'.

The diphthong counts err on the side of caution. The category of diphthong-bearing non-breaks is skewed somewhat by the high frequency of indefinite articles tagged with a full vowel, the canonical pronunciation: /eI/. Bearing this in mind, we re-calculated the value of chi-squared for files with non-significant correlations (*i.e:* A01, A03, A06, A09, A10), subtracting occurrences of /a:eI/ from the count for diphthong-bearing non-breaks and adding them to the non-diphthong-bearing non-breaks group. This made no difference to the result for each sub-file in all but one case: for A09, with 18 occurrences of /a:eI/, the re-calculated value of χ^2 is 8.579, with a two-tailed p-value of 0.0034.

Finally, calculating the chi-squared statistic for the correlation between diphthong-bearing words and breaks for the *whole* of Section A, we get a very significant result, for the data in Table 8.8: chi-squared equals 70.887 with one degrees of freedom and a two-tailed p-value which is less than 0.0001.

GROUPS	**OUTCOMES**		
	Breaks	Non-breaks	Totals
Diphthongs	550	1447	1997
No diphthongs	1075	4690	5765
Totals	1625	6137	7762

Table 8.8: A 2 x 2 contingency table records the observed frequency distribution for target groups and outcomes over all Section A files

8.9. Algorithm used in most recent dataset build

This section discusses the algorithm used to merge data from two different versions of the corpus (SEC and Aix-MARSEC) with canonical dictionary forms from ProPOSEL. A visual representation of the algorithm summarises preceding explanation and justification at each step in this segmented process (*cf.* Fig.8.1).

NLP resources at the University of Leeds include a version of SEC tagged with the Lancaster-Oslo-Bergen (LOB) tagset; but aligning word-LOB pairings in SEC with information from the current concatenated version of Aix-MARSEC (2006:02:27) was non-trivial. An initial problem is that some orthographic forms in SEC (*i.e.* hyphenated compounds and abbreviations) are decomposed into multiple phonetic and prosodic units in Aix-MARSEC: for example, the TextGrid file for A0802B in Aix shows decomposition of the word *x-ray* into two separate **n**arrow **r**hythm **u**nits (NRU), equivalent to two stressed feet.

SYLLABLES TIER: A0802B	**JASSEM TIER: A0802B**
8.3460000000000001	8.3460000000000001
""" e k s"	**"NRU"**
8.3460000000000001	8.3460000000000001
8.6959999999999997	8.6959999999999997
""" r eI"	**"NRU"**
8.6959999999999997	8.6959999999999997

Table 8.9: Data from 2 prosodic annotation tiers (syllables and rhythmic units) in an Aix-MARSEC TextGrid file

The first step was therefore to reconcile, manually, orthography in SEC Section A with that of Aix: for example, *TWA* (airlines) in A08 becomes *tee double u ay* and so on.

After automatically reconstituting enclitics in SEC (e.g. `will_MD not_XNOT` in LOB becomes `won’t_MD+XNOT`) in Step 2, the most intractable problem was mapping PoS tags from SEC with data from Aix (Step 3); in this merger, files are of different lengths, due to asynchronous distribution of punctuation (in SEC) and pauses/phrase break annotations (in Aix).

The dataset includes PoS tags from two schemes which differ in ‘delicacy’ (*cf.* Atwell, 2008): C5 is a much sparser tagset than LOB. It is also integral to dictionary lookup via ProPOSEL. The algorithm addresses this mismatch in delicacy between the tagsets in Steps 4 and 5. The former instantiates a live one-to-many mapping of C5<LOB PoS tags from the imported ProPOSEL lexicon. Examples in Table 8.10 show rafts of LOB tags mapped to C5 in the single category of adverbs, plus category combinations involving proper nouns, along with potential problems which lurk the other way: prepositions and subordinating conjunctions in LOB with more than one equivalent in C5.

Syntactic Category	**C5**	**LOB**
Adverbs	AV0	['QL', 'QLP', 'RB', 'RI', 'RBR', 'RBT', 'RN']
Enclitic: proper noun with *has*	NP0+POS	['NP$', 'NPL$', 'NPLS$', 'NPS$', 'NPT$', 'NPTS$']
Preposition: *of*	PRF	IN
Prepositions	PRP	IN
Subordinating	CJT	CS

conjunction: *that*		
Subordinating conjunctions	CJS	CS

Table 8.10: One-to-many mappings for C5 and LOB occur both ways

A match between LOB tokens in the merged dataset and the live mapping in ProPOSEL appends the corresponding C5 tag to dataset arrays (Step 5) and a patch is implemented to remove redundant C5 tags in cases of LOB<C5. Very few items remain untagged at this stage and can therefore be repaired manually: for example there were only 15 untagged items remaining out of 629 word tokens in Section A08.

Finally, ProPOSEL is transformed into a Python dictionary via its bespoke software tools (*cf.* Appendix 2), with compound (word + C5) keys mapped to prosodic-syntactic value arrays from selected fields in the lexicon. Intersection between dictionary keys and (word + C5) pairings in the dataset appends dictionary values to the parallel position in that sequence object (Step 6).

8.10 Language resources

The dataset built and used here for experimentation constitutes another language resource made available via this thesis: an open-source version of Section A (Commentary) in SEC, the ***S***poken ***E***nglish ***C***orpus (Taylor and Knowles, 1988) with multi-level parallel annotations juxtaposing linguistic information from different versions of the corpus with canonical dictionary forms, in a format optimized for query with Perl or Python and other text processing programs. This prototype prosody and POS annotated version of SEC (ProPOSEC) merges selected information from Aix-MARSEC (*i.e.* file number; word token; SAMPA phonetic transcription; and tonic stress marks assigned to each segment) with syntactic annotations from SEC, plus corresponding syntactic annotations and canonical pronunciations in the ProPOSEL lexicon. In addition, pauses denoting the original ‘gold-standard’ phrase break annotations in SEC are aligned with punctuation where appropriate.

Currently, the order and content of fields in the text file is as follows: (1) Aix-MARSEC file number; (2) word; (3) LOB PoS-tag; (4) C5 PoS-tag; (5) Aix SAM-PA phonetic transcription; (6) SAM-PA phonetic transcription from ProPOSEL; (7)

syllable count; (8) lexical stress pattern; (9) default content or function word tag; (10) DISC stressed and syllabified phonetic transcription; (11) alternative DISC representation, incorporating lexical stress pattern; (12) nested arrays of phonemes and tonic stress marks from Aix.

Listing 8.2 shows linguistic annotations in ProPOSEC for a prosodic-syntactic chunk initiated by a major clause boundary, the snippet *soon after it took off from Athens airport* from Section A08 of the corpus, with items in **bold** selected for further comment.

```
A0801|soon|RB|AV0|su:n|sun|1|1|C|'sun|'sun:1|[['s', 'u:', 'n'],
['\\', '\\', '\\']]

A0801|after|CS|CJS|A:ft@|'Aft@R|2|10|F|'#f-t@R|'#f:1 t@R:0|[['A:',
'f', 't', '@'], ['0', '0', '0', '0']]

A0801|it|PP3|PNP|rIt|It|1|1|F|'It|'It:1|[['r', 'I', 't'], ['0',
'0', '0']]

A0801|took|VBD|VVD|tUk|tUk|1|1|C|'tUk|'tUk:1|[['t', 'U', 'k'],
['`', '`', '`']]

A0801|off|RP|AVP|Qf|0f|1|1|C|'Qf|'Qf:1|[['Q', 'f'], ['0', '0']]

A0801|from|IN|PRP|fr@m|fr0m|1|1|F|'frQm|'frQm:1|[['f', 'r', '@',
'm'], ['0', '0', '0', '0']]

A0801|athens|NP|NP0|{TInz|'&TInz|2|10|C|No value|No value|[['{',
'T', 'I', 'n', 'z'], ['*', '0', '0', '0', '0']]

A0801|airport|NN|NN1|e@pO:t|'e@pOt|2|10|C|'8-p$t|'8:1 p$t:0|[['e@',
'p', 'O:', 't'], ['`/', '0', '0', '0']]

A0801|PAUSE|,|,
```

Listing 8.2: Parallel linguistic annotations for each word token include a prototype mapping between phones and tonic stress marks

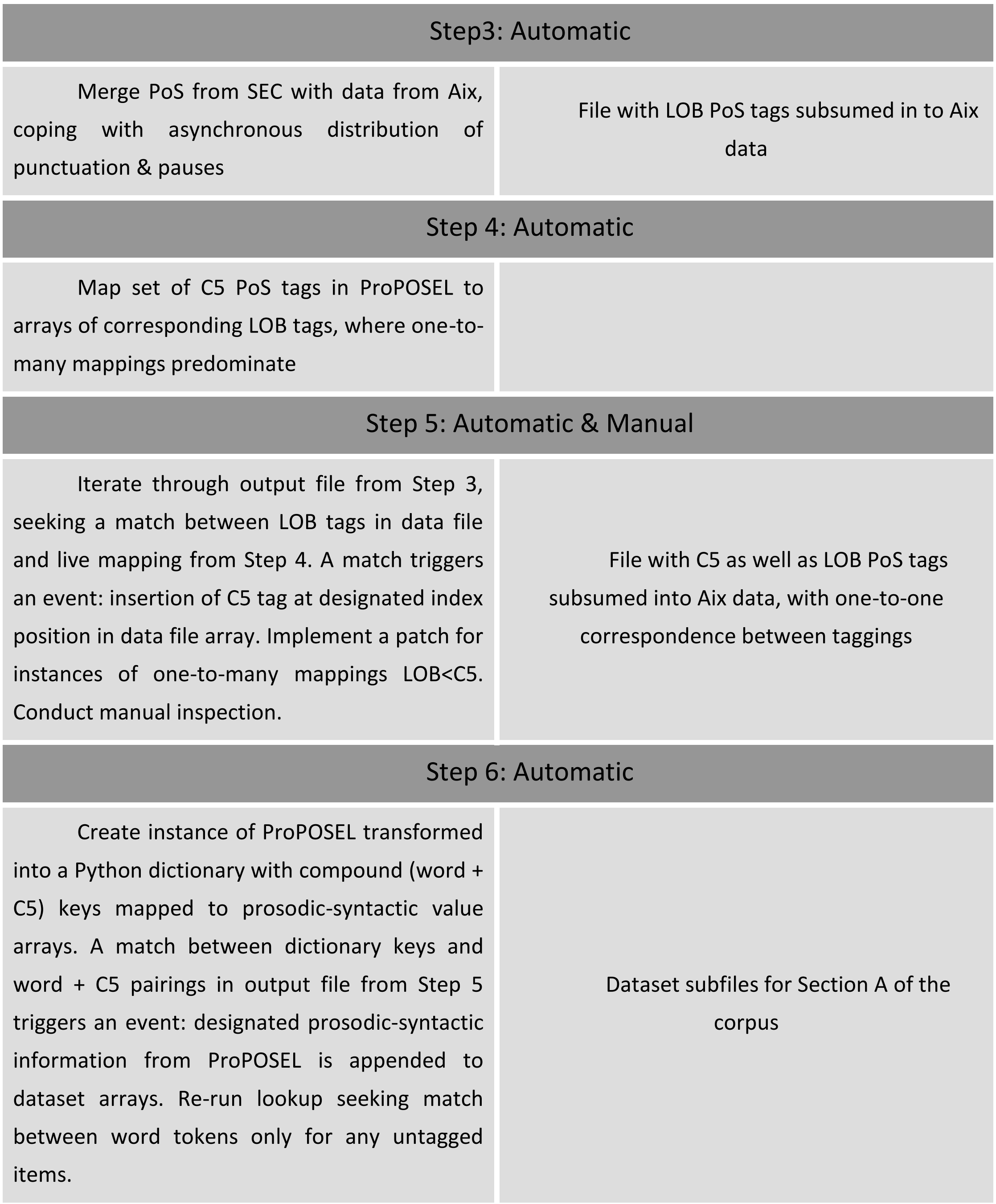
8.10.1 Elisions

Differences in ProPOSEC's SAM-PA transcriptions from Aix-MARSEC (field 5) and the lexicon (field 6) arise in part due to the former implementing elision rules for optimizing raw phonemic transcriptions (Auran *et al.*, 2004). Hence, in Listing 8.2, the Aix transcription for *it* shows a linking 'r'. Link-ups effected by w-glides and y-glides (Mortimer, 1985:46) are not included and constitute a potential enhancement for Aix-MARSEC and ProPOSEC. For example, greater verisimilitude to spoken English could be achieved quite simply by an extra rule governing use of the definite article (*cf.* 8.10.2).

8.10.2 Reduced forms

Another difference in ProPOSEC's SAM-PA transcriptions in fields (5) and (6) is more extensive representation of reduced vowels in function words in Aix-MARSEC. Hence we have an optimized versus canonical transcription for *from* in Listing 8.2. Definite articles in Aix-MARSEC are transcribed one of two ways: /D@/ - incorporating a schwa and identical to their SAM-PA transcriptions in the lexicon; and /DI/ - modelling coarticulation before vowels as in: /DI/ and /A:mI/ for *the army* (Aix-MARSEC A0402). As suggested in the previous section, elision prediction could include a linking 'y' in such instances: / DIjA:mI/ for *the ↺ army*.

Step 1: Manual	
Reconcile orthography in SEC file with Aix	Amended version of SEC file
Step 2: Automatic	
Reconstitute enclitics in SEC; lower case all words	



Step3: Automatic

Merge PoS from SEC with data from Aix, coping with asynchronous distribution of punctuation & pauses

File with LOB PoS tags subsumed in to Aix data

Step 4: Automatic

Map set of C5 PoS tags in ProPOSEL to arrays of corresponding LOB tags, where one-to-many mappings predominate

Step 5: Automatic & Manual

Iterate through output file from Step 3, seeking a match between LOB tags in data file and live mapping from Step 4. A match triggers an event: insertion of C5 tag at designated index position in data file array. Implement a patch for instances of one-to-many mappings LOB<C5. Conduct manual inspection.

File with C5 as well as LOB PoS tags subsumed into Aix data, with one-to-one correspondence between taggings

Step 6: Automatic

Create instance of ProPOSEL transformed into a Python dictionary with compound (word + C5) keys mapped to prosodic-syntactic value arrays. A match between dictionary keys and word + C5 pairings in output file from Step 5 triggers an event: designated prosodic-syntactic information from ProPOSEL is appended to dataset arrays. Re-run lookup seeking match between word tokens only for any untagged items.

Dataset subfiles for Section A of the corpus

Figure 8.1: Stages in dataset build

Chapter 9
ProPOSEC Dataset Transformation and Derivation of Non-traditional Prosodic Features for Supervised Machine Learning in WEKA

9.1. Overview

The previous two chapters have presented empirical evidence of a significant correlation in English between ‘gold-standard’ phrase break annotations in different varieties of spoken English and words containing complex vowels in their canonical dictionary pronunciations. Multi-level parallel annotations in the ProPOSEC dataset (*cf.* 8.10) facilitate statistical analyses of this kind.

The ProPOSEC dataset assembles a syntactic, rhythmic, and phonetic profile for each word in the corpus. However, converting this raw data into feature vectors for phrase break prediction using a machine learning toolkit such as WEKA (Hall *et al.*, 2009) is challenging for a number of reasons. One problem is the potential number of values for each attribute (*e.g.* the number of PoS in the tag set and the range of lexical stress patterns). Added to this is the problem of incorporating sufficient context into the language model: for example, the researcher may be interested in a window of *N* words either side of a given index position.

The focus for experimental work here, and in Chapters 7 and 8 of this thesis, is to gain insight into how interrelationships between syntax, rhythm and pronunciation might influence break placement. This chapter first describes how linguistic data arrays in the ProPOSEC dataset can be re-conceptualised as training instances for supervised machine learning via a knowledge engineering algorithm which represents each word token in the dataset as a vector of 31 nominal attribute-value pairings that complement traditional features (*i.e.* syntax and punctuation) with symbolic depictions of prosody. Findings from a series of boundary prediction experiments, with different combinations of traditional and non-traditional attributes and using the WEKA toolkit are then presented and discussed.

9.2. Data transformation

The raw data in Listing 9.1 shows selected linguistic annotations in ProPOSEC for a prosodic-syntactic chunk initiated by a major clause boundary, the string *soon after it took off from Athens airport* from Section A08 of the corpus. Only fields used in the algorithm are given and appear as follows: {word; C5 PoS-tag; lexical stress pattern; default content or function word tag; DISC stressed and syllabified phonetic transcription}. As a reminder of some terminology, lexical stress patterns are abstract representations of rhythmic structure, as in the sequence 201 for *disappear*, where each syllable is assigned a stress weighting: 1 for primary stress, 2 for secondary stress and 0 for unstressed elements. DISC phonetic transcriptions are unique in providing a one-to-one mapping between character and sound for long vowels, diphthongs and triphthongs, and affricates.

```
PAUSE|,|,
soon|AV0|1|C|'sun
after|CJS|10|F|'#f-t@R
it|PNP|1|F|'It
took|VVD|1|C|'tUk
off|AVP|1|C|'Qf
from|PRP|1|F|'frQm
athens|NP0|10|C|No value
airport|NN1|10|C|'8-p$t
PAUSE|,|,
```

Listing 9.1: Example of raw data in ProPOSEC showing word, C5 PoS tag, lexical stress pattern, content/function word tag, and stressed and syllabified DISC phonetic transcription

9.3. Rationale for attribute-value sets in re-conceptualised data

We are interested in: (i) uncovering prosodic information in plain text; (ii) how best to formulate categorical or descriptive representations of prosodic phenomena; and (iii) how ensuing features may improve automatic phrase break classification. Previous chapters present the solution to the first objective: simulating human reader and speaker performance by projecting prosody onto text via the ProPOSEL lexicon, a customised text annotation and text analytics tool. The focus here is on the second and third objectives, starting with the feature set, and the knowledge engineering algorithm used to derive values from raw annotations (as in Listing 9.1) and assign these categorical descriptors to each token. A visual summary of this algorithm is given in Fig. 9.1.

9.4. Reducing the POS-tag set for phrase break prediction

Taylor and Black's landmark comparative study of probabilistic and deterministic phrase break models over six experimental settings achieves a best score of 79% breaks-correct with a high order n-gram model and a reduced POS-tag set of 23 (*cf.* 3.3). Building on this cue, Read and Cox (2007) present a tagset reduction algorithm whose output of between 7 and 8 symbols is used to inform the feature set of their best performing model. Interestingly, existential *there* proves a useful predictor (*cf.* 4.3.2).

A knowledge engineering approach has here been used to economise on POS tags while remaining sensitive to the predictive potential of major clause markers, plus subtleties such as emphatic tendency (and hence *beat*) and prosodic coherence in, respectively, words classed in LOB as determiner/pronouns {such; all; both; same; few} and encliticised auxiliaries and modals {it'd; won't}. Under some schemes, notably CFP algorithms (*cf.* Busser *et al.*, 2001), such words would be classed as function words, hence initiating prosodic phrases and attracting false positive boundary placement. There are also deviant behaviours within syntactic categories that can be exploited, such as the relative dominance in noun attachment rate of the preposition *of* compared with other prepositions (Volk, 2006). The C5 tagset isolates both the preposition *of* and the conjunction *that* from like parts-of-speech and these distinctions have been preserved.

The POS attribute here is based on the traditional 8 parts of speech {nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, interjections} but expands some classes where differentiation is thought to influence boundary placement (*e.g.* Liberman and Church construe *object* pronouns as content words, unlike other pronouns) and adds a miscellaneous category for the infinitive marker *to*, foreign words, and items not to be tagged. The final set of values amounts to 22, and comprises: nouns; adverbs; 5 expanded classes: (i) verbs {`verb`, `modal`, `modal_negative`, `auxiliary`, `auxiliary_negative`}; (ii) adjectives {`adjective`, `article`, `determiner`}; (iii) pronouns {`pronoun`, `pronoun_reflexive`, `pronoun_object`, `pronoun_indefinite`, `pronoun_WH`};

(iv) prepositions {`preposition_of, all other prepositions`}; (v) conjunctions {`conjunction_that, all other conjunctions`}; interjections; a miscellaneous category; and a null value.

9.5. Prosodic attribute-value sets

In this study, each word has been classified in terms of 4 prosodic attributes: whether or not a word carries a beat; whether or not a word begins with a stressed syllable; whether or not a word ends with a stressed syllable, and if not, then the number of terminating *un*stressed syllables; whether or not a word contains a complex vowel. Each of these attributes is derived algorithmically by manipulating conditions based on projected prosody annotations from ProPOSEL and an important first step in the sequence (*cf.* Fig. 9.1) is ascertaining beat status for a given word.

9.5.1. Beat status

Lexical stress patterns and content-function word tags are mainly used to determine whether or not a word in context retains its canonical beat, based on the following assumptions. First, it is assumed all content words carry a beat, unless they are monosyllabic particles as in the phrase *get a move on*, where syntactic attachment of particle to preceding noun is, in a sense, reinforced prosodically via link-up (Mortimer, 1985), such that *move⤴on* becomes a single unit, with a hypothetical lexical stress pattern of `10` (*i.e.* stressed syllable followed by unstressed syllable). Thus, words with dual functionality as particles and prepositions are recognised, simultaneously, as syntactically distinct but prosodically analogous. Next, beat retention is assumed for numbers, reinforcement words (*cf.* 9.4), negative enclitics, function words and encliticised forms of more than one syllable, and any word for which lexicon look-up has failed, as these are largely proper nouns, and hence content words. Finally, it is assumed all other words forfeit their canonical beat in practice.

9.5.2. Jassem tag

The Aix-MARSEC corpus project includes rhythmic annotation tiers after Abercrombie versus Jassem (*cf.* Bouzon and Hirst, 2004), where stress feet are somewhat differently construed. For both theorists, stress feet in English begin with a stressed syllable; then, for the former, the foot continues *across* word boundaries if need be, until the next stressed syllable; while for the latter, the foot is decomposed into a narrow rhythm unit, terminating at a word boundary, and an anacrusis, denoting the gap or lull between that terminus and the next stressed syllable. The algorithm here uses categorical labels after Jassem because of the neat correspondence between concept and token, but distinguishes between beat-retaining words that begin with a stressed syllable and ones that do not. Thus in the fragment *before the hijacking of the* from A08, the noun *hijacking,* with primary stress on the first syllable, is tagged `NRU` for narrow rhythm unit, the preposition *of* is tagged `ANA` for anacrusis, and the preposition *before*, which carries a full beat on the second syllable, has a new combination tag of `ANA+NRU`, describing its rhythmical profile. Table 9.1 is a comparative breakdown of the given fragment according to the rhythmic scheme developed in this thesis and incorporating coarse-grained syntactic categorisation into content-function word groups and a binary flag for beat status.

Syntax	F	F	C	F	F
Tokens	before	the	hijacking	of	the
Beat flag	yes	no	yes	no	no
Jassem tag	ANA+NRU	ANA	NRU	ANA	ANA

Table 9.1: Descriptive classification of syntax and rhythm

9.5.3. Lexical stress pattern

The set of values for the attribute `lexical stress pattern` was considered to be too large (*e.g.* there are 124 patterns in ProPOSEL) and therefore this was rationalised down via a function which uses raw lexical stress patterns and beat assignment to determine whether a word ends with a stressed syllable; whether it ends with one, two, or multiple unstressed syllables; and whether lexical stress can be discounted (*i.e.* the word is a monosyllabic function word) or whether the value is simply missing.

9.5.4. Complex vowels

Complex vowels are the set of diphthongs and triphthongs in present day English and have been discussed in some detail in Chapter 7 of this thesis (*cf.* §7.4; 7.7; 7.7.1). DISC phonetic transcriptions employ a single character for each phonological segment. Complex vowels are represented by digits, (*cf.* 7.4), so a search was performed over DISC transcriptions to assign a binary tag for presence or absence of complex vowels for each token.

9.6. Punctuation

This thesis has already cited evidence from corpus-based studies and experimental work (§1.3) that punctuation is the single most important source of information for phrase break classification, finding approximately 50% of all breaks. We have also cited stylistic analyses where punctuation is used to trace pause patterns in Shakespearian blank verse (§7.5). Moreover, significance tests in this thesis consider *all* forms of punctuation (plus line terminals) as boundary annotations in Milton's verse (§7.5). For machine learning experiments in this chapter, we use three values to represent punctuation: words with attendant punctuation are classified as stops or medials, depending on punctuation type, and words with no attendant punctuation are labelled as non-terminals. This feature therefore implements the recommendation from Taylor (1996, p.145) that information about punctuation *type* should be used in speech synthesis systems.

9.7. Training instances

Taylor and Black classify junctures (whitespaces) as breaks or non-breaks and model prior probability of juncture type via a trigram context of two POS before and one after the juncture to be classified; while Busser *et al* use an extended fixed-width feature vector of two POS to both left and right of the focus position (*cf.* 3.3). One of the longer term goals following on from this thesis is to derive boundary-endorsing rhythmic templates from annotated speech corpora and from literary corpora, *especially* poetry (*cf.* examples of accented and de-accented midline stops in Milton’s blank verse in Table 7.3), and so training instances in this study capture prosodic-syntactic attribute-values for overlapping windows of five words. The class attribute then denotes break status for the third word only: words classed as breaks immediately precede boundary annotations in the corpus and are embedded and viewed within a context of two words both before and after, with dummy tokens inserted to supplement instances involving words at the beginning and end of each of the ten separate text files. Table 9.2 represents the complete training instance for the word *took* in the phrase *after it took off from*; the shaded row is a non-break and this determines the class attribute for this particular example.

Word	POS	Punctuation	Lexical Stress	Beat	Jassem	CV
after	conjunction	nonterminal	endsSingleUnstressedSyll	yes	NRU	no
it	pronoun	nonterminal	discountLexicalStress	no	ANA	no
took	**verb**	**nonterminal**	**endsStressedSyll**	**yes**	**NRU**	**no**
off	particle	nonterminal	discountLexicalStress	no	ANA	no
from	preposition	nonterminal	discountLexicalStress	no	ANA	no

Table 9.2: Non-break classification of training instance, where word classified is centrally embedded in N-gram of 5 tokens

The training instance itself takes the form:

> **took,** conjunction, nonterminal, endsSingleUnstressedSyllable, yes, NRU, no, pronoun, nonterminal, discountLexcialStress, no, ANA, no, verb, nonterminal, endsStressedSyllable, yes, NRU, no, particle, nonterminal, discountLexicalStress, no, ANA, no, preposition, nonterminal, discountLexicalStress, no, ANA, no, **nonBreak**

9.8. Abstract modelling of training instances

The supervised machine learning experiments in this chapter use features, or observations over a set of strings, to predict phrase breaks, encoded as a list of comma-separated values in an ARFF or CSV format. Another way of describing this abstract model is in terms of a mathematical equation analogous to the Drake Equation (*cf.* SETI, 2011) which explicitly lists the factors involved in predicting the number of technologically advanced civilisations that might exist in our galaxy. This way of stating a model lists all the features as factors in the equation:

```
PB = f (        W,
                POSn-2, PUNn-2, LSn-2, Bn-2, Jn-2, CVn-2,
                POSn-1, PUNn-1, LSn-1, Bn-1, Jn-1, CVn-1,
                POSn-2, PUNn, LSn, Bn, Jn, CVn,
                POSn+1, PUNn+1, LSn+1, Bn+1, Jn+1, CVn+1,
                POSn+2, PUNn+2, LSn+2, Bn+2, Jn+2, CVn+2 )
```

i.e. `PB` (phrase break) is some function of:

`W` = current word,

`POSn-2` = part-of-speech at position n-2,

`PUNn-2` = punctuation at position n-2,

`LSn-2` = lexical stress pattern at position n-2,

`Bn-2` = beat at position n-2,

`Jn-2` = Jassem at position n-2,

`CVn-2` = complex vowel at position n-2

… and so on

9.8.1. Weighted factors

In many of the experiments (*i.e.* Runs 8-33), we have set some of these factors to zero to see what effect this has on predictive power. For example, in the best-performing prosody-syntax model (Run 28, Table 9.5), there are only *nine* non-zero-weighted features. Moreover, this model also shows the dominance of syntax in the features used: *five* out of nine non-zero-weighted features concern part-of-speech.

`PB` (phrase break) is a function of:

`W` = current word,

`POSn-2` = part-of-speech at position n-2,

`POSn-1` = part-of-speech at position n-1,

`POSn` = part-of-speech at position n,

`POSn+1` = part-of-speech at position n+1,

`POSn+2` = part-of-speech at position n+2,

`LSn-2` = lexical stress pattern at position n-2,

`CVn-2` = complex vowel at position n-2,

`Jn+2` = Jassem at position n+2,

`CVn+2` = complex vowel at position n+2

It is also worth pointing out that this result is based purely on well-formed utterances (standardised and grammatically correct English usage), namely BBC radio broadcast news commentary. Thus it is not surprising that syntax is a good phrase break predictor here. Ideally, a fairer test of the contribution of symbolic prosodic features should be with less well-formed English, for example spontaneous conversational speech, or surreptitious recordings, or less "expert-crafted" text such as verbal autopsy reports (**Danso *et al.*, 2011**). However, none of these are available with mark-up for machine learning experiments; so the following sections necessarily report only on this “conservative” genre.

9.9. Overview of test procedure

The WEKA toolkit (Hall *et al.*, 2009) was used in a series of systematic boundary prediction experiments using different combinations of attributes (*i.e.* graphemic, syntactic and prosodic) to address the following questions:

1. Using the full feature set (punctuation, syntax, and prosody), can we improve on baseline performance?
2. When punctuation as top performing feature is removed from the feature set, does the addition of all 4 symbolic prosodic features improve on the performance of a syntax-only model?
3. Does the addition of complex vowels as a stand-alone prosodic feature enhance the performance of a syntax-only model?

These are the main questions; supplementary related questions are as follows:

4. Does the addition of other “stand-alone” prosodic features enhance the performance of a syntax-only model?
5. Does selectional inclusion of prosodic feature combinations enhance the performance of a syntax-only model?

These are supervised machine learning experiments which, because our dataset is small, use 10-fold cross-validation as test method. As is customary in phrase break prediction experiments (§1.5), we measure performance of the language model, *and this case the feature set*, via the number of major and minor boundary sites re-captured during test. Similarly, as in classic phrase break prediction experiments (Taylor and Black, 1998), we do not attempt to measure performance in terms of the model's ability to predict major as distinct from minor boundaries. Moreover, with respect to this decision, we have already presented evidence of: (i) transcriber differences in the assignment of boundary types in our dataset (§5.3; 5.3.1); and (ii) “inconsistencies” in boundary type assignment for one transcriber (§5.3.3). Hence in this study, we are only concerned with a two-class problem: predicting breaks (the minority class) or non-breaks (the majority class).

9.9.1. Testing via a range of classifiers

A decision was taken to use a range of generic classifiers during testing (*i.e.* decision trees: J48 and ADTree; rule learners: OneR and JRip; and Bayesian learners: BayesNet and AODE) to compare results from different learning schemes. This has proved advantageous in qualifying use of classification accuracy as sole evaluation metric, and in highlighting unequal classification error costs, where, for example, Bayesian learners capture more true positives but also generate more false positives than decision trees. These issues are discussed throughout Section 9.13. J48 is the main classifier used to address the research questions identified in Section 9.9 because it achieved the highest success rate in the first round of experiments (§Table 9.3). Finer points, such as exploring potential gains from sparse use of prosodic features, prompt comparative evaluation of J48 outputs versus at least one other classifer (§9.13).

9.9.2. Overview of evaluation metrics used

We have surveyed and summarised evaluation metrics commonly used in phrase break prediction experiments in Section 3.4. In machine learning, performance is traditionally measured first and foremost by *success rate* or *accuracy*: the total number of correct classifications for breaks plus non-breaks made during test vis-à-vis gold standard class labels for each test instance. We therefore use this metric but also juxtapose the accuracy measure with Balanced Classification Rate (BCR), namely the average of positive hits for each class, because class distributions in our dataset are skewed, such that an *un*intelligent classifier which labels each test instance with the dominant class label, achieves a respectable success rate of 79%. This additional BCR metric, and its implementation here, is more fully discussed in Section 9.13. We have also tabulated f-scores (§3.4) for the majority and minority class in each test run for comprehensive presentation, though these are not further discussed.

9.10. Test results with punctuation included as a feature

Table 9.3 summarises results in terms of 3 evaluation metrics from 10-fold cross validation tests on the transformed ProPOSEC dataset for experimental runs with punctuation included as a feature for various classifiers compared against two baselines: ZeroR and OneR. The former simply predicts the majority class (non-breaks in this case) for all instances; while the latter selects the best performing attribute for classification. Not surprisingly, this turns out to be punctuation. The simple ruleset:

```
{nonterminal -> nonbreak; terminal -> break; medial -> break}
```

results in correct classification of 6962 out of 7763 instances.

9.10.1. Discussion of test results with punctuation included as a feature

The first point to make with regard to Table 9.3 is that "unintelligent" baseline performance, namely the majority classifier, is reasonably high at 79.04% accuracy. The OneR classifier sets an even higher baseline success rate of 89.68%, demonstrating the fact that punctuation is a top-performing feature. Run 3, where J48 uses the full feature set of punctuation, syntax and all four additional symbolic prosodic features, improves on OneR baseline performance, with a success rate of 90.07%. This constitutes the best accuracy score achieved throughout this first series of tests, and also represents an improvement on OneR performance in terms of two other evaluation metrics: f-score for minority class (0.73 instead of 0.67); and Balanced Classification Rate, namely an average of the hits on each class (0.80 instead of 0.75). The improvement in accuracy turns out not to be statistically significant however. This is more fully discussed in Section 9.13, which also addresses the rationale for test runs with other classifiers in Table 9.3, and interesting insights gained in the process.

Total Number of Instances: 7763
Total Non-Breaks: 6136; Total Breaks: 1627
Prior probabilitiy Majority Class: 0.79; Prior Probability Minority Class: 0.21

Run	**Classifier**	**Number of features**	**Description of feature set**	**% Success rate**	**TP**	**FN**	**TN**	**FP**	**F-score: Majority Class**	**F-score: Minority Class**	**BCR: Balanced classification rate (Higher is better)**
1	ZeroR	31	Whole feature set	79.04	0	1627	6136	0	0.88	0	0.50
2	OneR	31	Whole feature set (rule=punct)	89.68	826	801	6136	0	0.94	0.67	0.75
3	J48	31	Whole feature set	90.07	1038	589	5954	182	0.94	0.73	0.80
4	JRip	31	Whole feature set	88.92	981	646	5922	214	0.93	0.70	0.78
5	ADTree	31	Whole feature set	90.03	1016	611	5973	163	0.94	0.72	0.80
6	AODE	31	Whole feature set	86.59	1342	285	5380	756	0.91	0.72	0.85
7	BayesNet	31	Whole feature set	82.67	1388	239	5030	1106	0.88	0.67	0.84

Table 9.3: Experimental runs for various classifiers versus baseline performance with punctuation included as a feature

9.11. Test results without punctuation included as a feature

The objective in the first round of tests was to see if, given a feature set augmented with symbolic prosodic features, our classifier (J48) could improve on baseline performance for phrase break prediction set by a majority (ZeroR) classifier and then a OneR classifier. An improvement was recorded in both cases; and since this classifier achieved the highest accuracy score, it was retained in subsequent tests. The objective in this second round of tests is to evaluate classifier performance when punctuation, the top-performing feature, is removed - and more particularly, to see if a language model using prosodic as well as syntactic features can improve on a syntax-only model. Experimental runs also record classifier performance versus the baseline and versus a syntax-only model when syntax is supplemented by each of the 4 prosodic features in turn, namely: (i) complex vowels; (ii) lexical stress or word-internal syllable weightings; (iii) beat; (iv) jassem or coarse-grained, word-internal rhythmic profile. From the point of view of many natural language engineering applications, it would be detrimental to remove punctuation as a feature. Taylor (1996, p.143) argues that '...all punctuation marks can contribute to the generation of tone group boundaries.' However, for the purposes of experimentation, and given that there are also applications with direct speech as input, and without punctuation mark-up, punctuation is now removed from the feature set.

9.11.1. Discussion of results where punctuation is removed as a feature

The ZeroR baseline is fixed at 79.04%. The baseline success rate for the syntax-only OneR classifier is 81.62%. Not surprisingly, the OneR classifier uses `postpos1` as sole discriminator, which can effectively be interpreted as the chink-chunk rule: the sequence open-class word + closed class word is an optimal phrase break location. However, since we are measuring whether additional prosodic information improves performance for phrase break prediction in the absence of punctuation, the figure to beat is J48's performance of 85.68% with syntactic features only. This figure is taken from Run 12 in Table 9.4. On this we get negative results as follows, though the differences do appear to be marginal, indicating that neither model is *significantly* better than the other: (i) 85.25% versus 85.68% when all prosodic features are selected in Run 13; (ii) 85.48% versus 85.68% when syntax is supplemented with complex vowels only in Run 16; (iii) 85.29% versus 85.68% when syntax is supplemented with lexical stress only in Run 17; and (iv) 85.59%

versus 85.68% when syntax is supplemented with either the beat or jassem feature in Runs 18 and 19 respectively.

Total Number of Instances: 7763
Total Non-Breaks: 6136; Total Breaks: 1627
Prior probabilitiy Majority Class: 0.79; Prior Probability Minority Class: 0.21

Run	Classifier	Number of features	Description of feature set	% Success rate	TP	FN	TN	FP	F-score: Majority Class	F-score: Minority Class	BCR: Balanced classification rate (Higher is better)
8	OneR	6	Syntax only (rule=postpos1)	81.62	330	1297	6006	130	0.89	0.32	0.59
9	OneR	26	Prosody-syntax (rule=postpos1)	81.62	330	1297	6006	130	0.89	0.32	0.59
10	OneR	21	Prosody only (rule=poststress1)	79.04	9	1618	6127	9	0.88	0.01	0.50
11	OneR	6	CV only (rule=postcv1)	79.17	10	1617	6136	0	0.88	0.01	0.50
12	J48	6	Syntax only	85.68	921	706	5730	406	0.91	0.62	0.75
13	J48	26	Prosody-syntax	85.25	870	757	5748	388	0.91	0.60	0.74
13a	ADTree	26	Prosody-syntax	85.25	1050	577	5568	568	0.91	0.65	0.78
14	J48	21	Prosody only	78.71	363	1264	5747	389	0.87	0.31	0.58
15	J48	6	CV only	79.04	0	1627	6136	0	0.88	0	0.50
16	J48	11	Syntax and CVs	85.48	904	723	5732	404	0.91	0.62	0.74
17	J48	11	Syntax and stress	85.29	905	722	5716	420	0.91	0.61	0.74
18	J48	11	Syntax and beat	85.59	903	724	5741	395	0.91	0.62	0.75
19	J48	11	Syntax and jassem (rhythm)	85.59	903	724	5741	395	0.91	0.62	0.75

Table 9.4: Experimental runs for J48, a generic decision tree classifier, versus OneR baseline performance minus punctuation as a feature. (§10.9.3) for comment on ADTree Run 13a.

9.12. Tests where syntax is supplemented by combinations of prosodic features

Since the addition of single prosodic features (Runs 16-19) gave a marginal improvement on use of all prosodic features (Run 9), a "trial and error" approach was adopted to see which combinations of prosodic features (if any) could improve on a syntax-only model. An abstraction of the best performing model in this experimental round has already been presented in Section 9.8.1. Table 9.5 compares performance of different prosodic-syntactic feature combinations with the accuracy rate achieved by the J48 syntax-only model: 85.68%. All prosodic-syntactic feature variations tabulated represent a marginal improvement on the syntax-only model. We focus on the best result in Run 28 (Table 9.5). This model has a success rate of 85.80% and uses complex vowels in 2 index positions (i - 2 and i + 2) either side of the index to be classified, plus word-internal rhythmical/syllabic information: lexical stress (i - 2) and jassem (i + 2).

Total Number of Instances: 7763

Total Non-Breaks: 6136; Total Breaks: 1627

Prior probabilitiy Majority Class: 0.79; Prior Probability Minority Class: 0.21

Run	Classifier	Number of features	Description of feature set	% Success rate	TP	FN	TN	FP	F-score: Majority Class	F-score: Minority Class	BCR: Balanced classification rate (Higher is better)
12	**J48**	**6**	**Syntax only**	**85.68**	**921**	**706**	**5730**	**406**	**0.91**	**0.62**	**0.75**
20	J48	8	Syntax(CV + Beat in Post2)	85.73	921	706	5734	402	0.91	0.62	0.75
21	J48	8	Syntax(CV + Jassem in Post2)	85.73	921	706	5734	402	0.91	0.62	0.75
22	J48	10	Syntax(All in Pre2)	85.73	929	698	5726	410	0.91	0.63	0.75
23	J48	7	Syntax(CV in Pre2)	85.73	926	701	5729	407	0.91	0.63	0.75
24	J48	7	Syntax(Stress in Pre2)	85.73	925	702	5730	406	0.91	0.63	0.75
25	J48	13	Syntax(All in Pre2; 3 in Post2)	85.71	940	687	5714	422	0.91	0.63	0.75
26	J48	9	Syntax(CVPre2; JassemCVPost2)	85.74	927	700	5729	407	0.91	0.63	0.75
27	J48	10	Syntax(StressCVPre2; BeatCVPost2)	85.77	935	692	5723	413	0.91	0.63	0.75
28	**J48**	**10**	**Syntax(StressCVPre2; JassemCVPost2)**	**85.80**	**937**	**690**	**5724**	**412**	**0.91**	**0.63**	**0.75**

Table 9.5: Experimental runs for J48 with syntax plus different combinations of prosodic features versus a syntax-only model

9.12.1. McNemar significance test

To determine whether or not the score of 85.80% (a prosody-syntax phrase break model) in Run 28 represents a *significant* improvement on 85.68% (a syntax only phrase break model) in Run 12, a further test was performed: McNemar's significance test for comparing the performance of two algorithms. The test requires classification data on matched pairs, and in this case, this is provided by output predictions for the same instances for the two different models in WEKA, as in Table 9.6.

Output Predictions in WEKA for syntax-only J48 model in Fold 1

```
770    2:break 1:nonbreak      +  *0.925  0.075
771    2:break 1:nonbreak      +  *0.726  0.274
772    2:break 1:nonbreak      +  *0.5    0.5
773    2:break 1:nonbreak      +  *0.842  0.158
774    2:break    2:break          0.193 *0.807
775    2:break    2:break          0.423 *0.577
776    2:break    2:break          0.193 *0.807
777    2:break 1:nonbreak      +  *0.667  0.333
```

Output Predictions in WEKA for syntax-prosody J48 model for same Fold 1

```
770    2:break 1:nonbreak      +  *0.783  0.217
771    2:break 1:nonbreak      +  *0.778  0.222
772    2:break    2:break          0.167 *0.833
773    2:break 1:nonbreak      +  *0.719  0.281
774    2:break    2:break          0.111 *0.889
775    2:break 1:nonbreak      +  *0.995  0.005
776    2:break 1:nonbreak      +  *0.832  0.168
777    2:break 1:nonbreak      +  *0.786  0.214
```

Table 9.6: Matched pairs of output predictions in WEKA for the same instances in the syntax-only versus syntax prosody models

Counts for concordant and discordant pairs were derived from such classification data in WEKA and assembled in a 2 x 2 contingency table as follows (Table 9.7).

	Syntax-only: `correct`	Syntax-only: `incorrect`
Prosody-syntax: `correct`	6621	**40**
Prosody-syntax: `incorrect`	**30**	1072

Table 9.7: Concordant and discordant results for the syntax-only and prosody-syntax models (Runs 12 and 28)

McNemar’s chi-squared significance test only considers whether or not there is a significant difference in proportions in the discordant pairs (shaded in Table 9.7). In this case, the difference is not significant: the two-tailed p-value is 0.28, and the odds ratio is 1.33 with a 95% confidence interval.

9.13. Do prosodic features add value?

Do prosodic features add value for phrase break prediction? This is a difficult question. On the one hand, there is *some* evidence to suggest that the addition of prosodic features *does* enhance performance depending on which model and which evaluation metric is used. This evidence comes initially from Table 9.3, which tabulates results for various classifiers with all features present: punctuation, syntax and prosody. While decision trees (J48 and ADTree) have the highest success rates, the Bayesian classifiers (AODE and BayesNet) clearly capture more true positives, suggesting these classifiers have learnt the concept associated with the minority class better than others, and more importantly, they exhibit better Balanced Classification Rates.

9.13.1. Balanced Classification Rate versus Accuracy

Before re-considering results from Table 9.3, plus additional results in Table 9.8, we might consider the argument for preferring BCR to classical overall accuracy, even though this metric has not been used in classic phrase break prediction experiments (*cf.* 3.4). In our experimental dataset, and in datasets for phrase break prediction in general, the classes are not evenly distributed: there is not a 50/50 chance that each whitespace between words can be classified as a break or non-break. Instead, datasets are imbalanced, leading to apparently “respectable” success rates for unintelligent ZeroR classification based on skewed class priors, in this case 79% (non-breaks) versus 21% (breaks). Accuracy or success rate does not consider these relative class distributions and unequal classification error costs, whereas BCR places equal emphasis on model capture of true positives (*sensitivity*) as well as true negatives (*specificity*), and is not pre-empted by an imbalanced dataset. Hence there are examples in machine learning literature of “success” for classification being interpreted as BCR rather than the traditional accuracy measure. For example, BCR is the preferred metric in the STAMINA (State Machine Inference Approaches) competition “…to drive the evaluation and improvement of software model-inference approaches…" (Walkinshaw *et al.*, 2010), and has been used for evaluation in machine learning and knowledge discovery (Helleputte and Dupont, 2009); and document analysis systems (Gazzah and Ben Amara, 2008). Table 9.8 subsumes Table 9.3, and juxtaposes runs from Table 9.4 with new material (Runs 29-33) from which to draw overall conclusions. It still gives BCR as:

0.5 * ((TP / total positive instances) + (TN / total negative instances))

However, it also presents a version of BCR known as *harmonic* BCR (Walkinshaw *et al.*, 2010). This is computed as:

2 * ((sensitivity * specificity) / (sensitivity * specificity))

where *sensitivity* equals: TP / (TP + FN), and where *specificity* equals: TN / (TN + FP). This is the metric used in STAMINA.

9.13.2. Verifying accuracy via significance testing with punctuation present

In Table 9.8, J48's success rate of 90.07% (Run 3) using the entire feature set does not represent a *significant* improvement on baseline performance, namely the 89.68% achieved by the OneR model using punctuation as sole predictor of phrase breaks. The same applies to AODE's success rate of 86.59% (Run 6) which, despite the high BCR metrics, shows a significantly *worse* result. Significant improvement (or otherwise) is here measured (Table 9.9) via the corrected re-sampled t-test, implemented in WEKA's *Experimenter*, and used to verify BCR (Nadeau and Bengio, 2003; Helleputte and Dupont, 2009). This finding has also been corroborated via the McMemar test (9.12.1) and the results presented in Tables 9.10 and 9.11. By the same token, AODE's punctuation and syntax model (Run 33) represents the *only* significant improvement on the baseline punctuation rule in this entire series of tests (Tables 9.8 and 9.13).

Total Number of Instances: 7763
Total Non-Breaks: 6136; Total Breaks: 1627
Prior probabilitiy Majority Class: 0.79; Prior Probability Minority Class: 0.21

Run	**Classifier**	**Number of features**	**Description of feature set**	**% Success rate**	**TP**	**FN**	**TN**	**FP**	**BCR: Balanced classification rate (Higher is better)**	**Harmonic BCR**
1	ZeroR	31	Whole feature set	79.04	0	1627	6136	0	0.50	0.00
2	OneR	31	Whole feature set (rule=punct)	89.68	826	801	6136	0	0.75	0.67
3	J48	31	Whole feature set	90.07	1038	589	5954	182	0.80	0.77
4	JRip	31	Whole feature set	88.92	981	646	5922	214	0.78	0.74
5	ADTree	31	Whole feature set	90.03	1016	611	5973	163	0.80	0.76
6	**AODE**	31	**Whole feature set**	**86.59**	1342	285	5380	756	**0.85**	**0.85**
7	BayesNet	31	Whole feature set	82.67	1388	239	5030	1106	0.84	0.84
8	OneR	6	Syntax only (rule=postpos1)	81.62	330	1297	6006	130	0.59	0.34
29	AODE	6	Syntax only	85.06	910	717	5693	443	0.74	0.70
9	OneR	26	Syntax and prosody (rule=postpos1)	81.62	330	1297	6006	130	0.59	0.34
30	AODE	26	Syntax and prosody only	81.13	1162	465	5136	1000	0.74	0.77
11	OneR	11	Syntax and CVs (rule=postpos1)	81.62	330	1297	6006	130	0.59	0.34
31	**AODE**	11	**Syntax and CVs only**	**84.63**	917	710	5653	483	0.74	0.70
32	J48	11	Syntax and punctuation	89.66	857	770	6103	33	0.76	0.69
33	**AODE**	11	**Syntax and punctuation**	**90.35**	1159	468	5855	281	**0.83**	**0.82**

Table 9.8: Comparative performance of classifiers in terms of Accuracy, Balanced Classification Rate (BCR), and Harmonic BCR

```
Analysing:  Percent_correct
Datasets:   1
Resultsets: 2
Confidence: 0.05 (two tailed)
Date:       19/07/11 09:00


Dataset                    (1) rules.On | (2) trees
                          --------------------------
wekadataCV_ALL.txt        (100)   89.68 |   89.74
                          --------------------------
                               (v/ /*) |   (0/1/0)
Skipped:

Key:

(1) rules.OneR '-B 6' 3010129309850089072
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
```

```
Datasets:   1
Resultsets: 2
Confidence: 0.05 (two tailed)
Date:       19/07/11 08:54


Dataset                    (1) rules.On | (2) bayes
                          --------------------------
wekadataCV_ALL.txt        (100)   89.68 |   86.51 *
                          --------------------------
                               (v/ /*) |   (0/0/1)
Skipped:

Key:

(1) rules.OneR '-B 6' 3010129309850089072
(2) bayes.AODE '\"-F \" 0' 9197439980415113523
```

Table 9.9: Results for corrected re-sampled t-test in WEKA Experimenter for J48 and AODE compared against the OneR baseline.

	OneR: correct	OneR: incorrect
J48: correct	6780	**212**
J48: incorrect	**182**	589
2-tailed p-value = 0.14; odds ratio = 1.17 with 95% confidence		

Table 9.10: Concordant and discordant results for J48 using all available features compared against OneR (punctuation) as control in McNemar's test

	OneR: `correct`	OneR: `incorrect`
AODE: `correct`	6190	**532**
AODE: `incorrect`	**772**	269
`2-tailed p-value = 0.0001; odds ratio = 0.69 with 95% confidence`		

Table 9.11: Concordant and discordant results for AODE using all available features compared against OneR (punctuation) as control in McNemar’s test. Here, OneR performs significantly *better* than AODE.

	OneR: `correct`	OneR: `incorrect`
AODE: `correct`	6680	**334**
AODE: `incorrect`	**282**	467
`2-tailed p-value = 0.04; odds ratio = 1.18 with 95% confidence`		

Table 9.12: Concordant and discordant results for AODE using punctuation and syntax compared against OneR (punctuation) as control in McNemar’s test

9.13.3. Verifying accuracy via significance testing with punctuation absent

Table 9.8 summarises experimental permutations for different classifiers with and without punctuation; this summarisation is legitimate given the preceding systematic presentation of results for J48 as background. Run 31 in Table 9.8 records performance for an AODE model *minus* punctuation and using syntax and complex vowels as features. The success rate of 84.63% is an interesting result as it represents a *significant* improvement on 81.62% for the OneR syntax-only model (Run 8) which uses `postpos1` as phrase break indicator: effectively, the chink-chunk rule. Again, this is verified via McNemar’s test (Table 9.13). By the same token, however, AODE’s syntax-only performance (Run 29), also represents a significant improvement on the baseline and attains a higher success rate: 85.06% versus 84.63%. Here significance is assumed, given the finding in Table 9.13.

	OneR: correct	OneR: incorrect
AODE: correct	5923	**647**
AODE: incorrect	**413**	780
2-tailed p-value = 0.0001; odds ratio = 1.57 with 95% confidence		

Table 9.13: Concordant and discordant results for AODE using syntax and complex vowels compared against OneR (postpos1) as control in McNemar's test

9.14. Concluding remarks

From the evidence presented in this chapter, based solely on a limited dataset of grammatically well-formed, expertly-crafted BBC radio broadcast transcripts, it appears that symbolic prosodic features do not improve phrase break prediction over syntax-only predictors, and that this finding applies both when punctuation is present, and when it is withheld as a feature. In fact it is fair to say that at *best*, the inclusion of prosodic features has not achieved *significant* gains in performance (Run 3 versus Run 2), and that at *worst*, prosody has detracted from performance by introducing "noise" (Run 6 versus Run 33). Less prosody means less noise, and that would explain why *single* prosodic features (e.g. the complex vowels in Run 31, Table 9.8), or *sparser combinations* of prosodic features (e.g. Run 28, Table 9.5), get better results. Hence, these findings all tend to *dis*prove the hypothesis that symbolic prosodic features, in addition to syntax and punctuation, will enhance performance in phrase break prediction. This finding is more fully discussed in the main Conclusions chapter.

Step 1: Automatic	
Use PAUSE tokens to classify preceding words as breaks (or non-breaks if there is no subsequent PAUSE).	Attribute: class Values: 2 + NONE
Step 2: Automatic	
Implement a “rationalise POS” function via C5 tags to reduce the number of values for syntax as an attribute, while preserving distinctive categories of interest.	Attribute: POS Values: 21 + NONE
Step3: Automatic	
Use (mainly) lexical stress patterns and content-function word status to determine whether or not a word in context retains its canonical beat.	Attribute: beat Values: 2 + NONE
Step 4: Automatic	
Use beat assignments from the previous step, together with lexical stress patterns, to assign Jassem tags, indicating whether or not a word begins with a stressed syllable, and if not, whether a stressed syllable occurs before the word boundary.	Attribute: Jassem Values: 3 + NONE
Step 5: Automatic	
Implement a “rationalise lexical stress” function to reduce the value set of lexical stress patterns.	Attribute: rhythmic profile Values: 6 + NONE
Step 6: Automatic	
Generate a punctuation attribute via presence or absence of punctuation, plus *type* of punctuation (*i.e.* medial or terminal).	Attribute: punctuation Values: 3 + NONE
Step 7: Automatic	
Use DISC phonetic transcriptions to determine whether or not a word contains a diphthong or a triphthong.	Attribute: complex vowel Values: 2 + NONE

Steps 8 (Automatic), 9 (Manual) and 10 (Automatic)	
Generate sliding windows of length 5 to embed and contextualise the index to be re-classified (*i.e.* the middle index) during testing.	Training instances of 31 attribute-value pairings for: index – 2, index – 1, index, index + 1, index + 2, and the class attribute.
Add dummy leading and trailing indices to embed and contextualise words at beginnings and ends of ProPOSEC text files.	
Print n-gram instances to file, removing the class attribute for all but the middle index, and appending this attribute at the end of each instance.	

Figure 9.1: Summary of stages in algorithmic transformation of ProPOSEC data into a training set for machine learning

Chapter 10
Thesis Summary, Conclusions, and Ideas for Further Work

10.1 Overview

This final chapter first reviews the main threads running through this thesis, and then draws conclusions from empirical work about: (i) the correlation of *descriptive* as opposed to *acoustic* prosodic phenomena – and in particular complex vowels – and boundary annotations in different texts and different spoken genres; and (ii) the predictive potential of such symbolic prosodic features for the machine learning task of phrase break prediction. Recommendations for future work include/cover: (i) extension and application of the ProPOSEL lexicon as speech-to-viseme generator for avatar creation; and (ii) application of text analytics techniques for *English* developed in this thesis to explore phrasing strategies in *Arabic*, another stress-timed language. Finally, the chapter summarises PhD impact, originality, and contribution to research field.

10.2. Thesis main threads: shallow parsing and prosodic phrasing

Prosodic chunking is a language universal and there is acceptance that such phrasing is simpler, shallower and flatter than syntactic structure, hence the tradition of robust shallow parsing for predicting prosodic phrase boundaries in unseen text, with CFP algorithms implemented at Bell (Abney, 1994) and Toshiba (Knill, 2009). A comprehensive study (Ingulfsen *et al.*, 2005) found that shallow parse features at different levels of granularity, complemented by information about strong/weak syntactic coupling from Link grammar, achieved a good balance between the IR metrics of precision and recall for phrase break prediction. This study, along with Taylor and Black (1998), also recognises punctuation as a top performing feature for this task, re-invoking the status of punctuation as prosodic annotation:

> ‘...Elizabethan popular writers...wrote for a people whose social intercourse had developed the art of conversation – their punctuation in this connection is highly suggestive, and so is the size of their vocabulary...’ (Leavis, 1965, p. 88).

10.2.1. Ambiguity of function words

In Chapter 4 of this thesis, the focus of early work shifts from attempts to define syntactic sequences equivalent to prosodic phrases for shallow parsing (*i.e. contents* of prosodic phrases), to PoS either side of the boundary position itself. An improved f-score was achieved via a prototype two-stage chunker, where rule-ordering constitutes both a linguistic and algorithmic challenge. Chapters 4 and 5 question the binary divide into content and function words for CFP rules: 'gold standard' CF allocation of blended categories, most notably particles and prepositions, is variable. Moreover, some function words are bi-syllabic and by virtue of this fact, their rhythmic constitution is analogous to bi-syllabic content words and may influence boundary placement. Finally, there is as yet no agreed set of reliable content-function word defaults for the phrase break prediction task.

10.3. Thesis main threads: the variability of prosody

As well as revisiting the theme of blended categories and dual-functioning PoS, Chapter 5 cites Taylor and Black (1998) and Atterer and Klein (2004) who question the validity of evaluating prosodic phrasing models in terms of one corpus template. Many insertion or deletion errors for boundary prediction are not errors at all but legitimate, variant phrasing strategies. Again, these alternative phrasings are embedded in the text, as is the case with variant parsing strategies.

Chapter 5 suggests a follow-up project where a tried and tested prosodic phrasing model is used to generate alternative prosodies for each sentence in the corpus which would then be subjected to human judgement to ascertain accuracy and naturalness and finally incorporated into a parallel corpus. It would not be the first time that variant annotations have been included in the gold standard. MultiTreebanks have been used for comparative analyses of rival parsing programs; for prosody, and because of its inherent variability, parallel prosodic realisations of each sentence in the corpus would facilitate more robust, noise-tolerant evaluation of phrasing models.

10.3.1. The distinction between chunking and highlighting

Another idea emerging from Chapter 5 of this thesis is different kinds of boundaries categorised as chunkers and highlighters. The former may co-occur with major clause markers but may also be manifest at other levels in the syntax tree. The latter occur much lower down the tree; and in some cases, speakers may choose to relinquish an obvious chunking boundary in favour of more idiosyncratic highlighting in order to emphasise certain structures or constituents: speech acts or semantic roles.

10.4. Thesis main threads: projecting prosody onto text via ProPOSEL

Chapter 6 reports on the trend towards leveraging real-world linguistic knowledge to enhance performance in machine learning for language engineering tasks (*e.g.* Furui, 2009), and highlights a deficiency of *a priori* knowledge of prosody in phrasing models for English. Furthermore, it concurs with studies that recognise how human readers project their internalised knowledge of prosody onto text even during silent reading to inform parsing and understanding. The prosody inherent in text, and currently absent in learning paradigms for phrase break models, is revealed in the multiple annotation tiers in Aix-MARSEC. The ProPOSEL lexicon developed as part of this thesis is a linguistic repository from which to extract real-world knowledge of prosody and syntax for projection (*i.e.* annotation) onto any text; it is possible, for example, to regenerate symbolic rhythmic labels such as Aix-MARSEC’s *narrow rhythm units* and *anacruses* from lexical stress patterns in ProPOSEL.

10.4.1. Prosodic information in ProPOSEL

Although ProPOSEL does not incorporate variant pronunciations, it does contain stressed and syllabified phonetic transcriptions. Pronunciation lexica for ASR often record a range of linear phonetic transcriptions for a given word but a transcription string also needs to be given structure (syllabification) and depth (stress weigthings) which together define the rhythmic profile or lexical stress pattern for a word and are subsisting attributes of a word. Chapter 6 of this thesis has uncovered variance in pronunciation lexica with regard to syllabification and secondary stress assignment. For lexical items where there is no ambiguity in terms of syllable count, the lexical stress pattern tells us something more fundamental about a word: its rhythmic structure. The potential of this feature for predicting rhythmic juncture is explored in machine learning experiments in Chapter 9.

10.5. Thesis main threads: the ProPOSEL lexicon as model

There is a perceived need for fine-grained syntactic, morphological and phonetic information in lexica designed for language engineering tasks such as TTS, ASR and Machine Translation. This thesis argues the case for discriminating word class information in lexica designed for linkage with corpora. The ProPOSEL model maps between four syntactic annotation schemes, with software tools for refining this mapping and re-tagging a corpus annotated with one of its hosted tag-sets. The contributing lexical resources forming the basis of ProPOSEL – OALD, CUVPlus, BNC, CELEX, CMU, Penn Treebank and LOB – have each been used in a variety of research projects covering psycholinguistics, language engineering and corpus linguistics. Hence, in combining lexical information from all these resources, ProPOSEL is applicable in all these research areas, and many more.

By integrating a range of different resources, and enabling a variety of access strategies (*cf.* 6.8), with consultation based on various combinations of partial syntactic and prosodic knowledge of target words, ProPOSEL represents groundwork for the next generation of electronic dictionaries.

10.5.1. Cognitive aspects of the lexicon

Phonology fields in ProPOSEL constitute a range of access routes for users and enable lookup via sound, syllables, and rhythmic structure as alternatives to orthographic form. Human users of electronic dictionaries can start from partial concepts or patterns when they are generating a message or looking for a (target) word. Conceptual inputs of dictionary users may be based on semantic cues, such as conceptual primitives, semantically related words, partial definitions (e.g. synsets); but speakers/writers may also be searching for a word which matches syntactic, phonetic or prosodic partial patterns, for example, seeking a matching rhythm or rhyme. While meaning is clearly the focus of many lexicography researchers, access by sound, rhythm and prosody, plus syntactic similarity, may also prove useful complementary strategies for some users.

Another key issue for lexicography (*cf.* Workshop on Cognitive Aspects of the Lexicon, Coling 2008) is robust, yet flexible organisation of resources. By building on and integrating with Python and NLTK, ProPOSEL can be accessed by other NLP tools or via the standard Python interface for direct browsing and search. ProPOSEL is also a potential exemplar for lexical entry standardisation. Many lexicographers focus on standardisation of semantics or definitions; but standardisation of syntactic, phonetic and prosodic information is also an issue. The pragmatic approach of this thesis is to integrate lexical entries from a range of resources into a standardised Python dictionary format, where the dictionary is reconceived and dynamically reconstituted as an associative array. Users can thus manipulate the text file to perform filtered searches on subsets of the lexicon and access wordforms via sound, syllables and rhythmic structure.

10.6. Thesis main hypothesis: development

The design (*e.g.* feature selection) and evaluation of language models for automatic phrase break prediction is challenged by the inherent variance of prosody itself. Prosodic-syntactic chunking is a language universal, however, and the survey of phrase break models for English (*cf.* Chapter 3) confirms that syntax is integral to the task. Syntactic features may be shallow (*e.g.* PoS-tags or the content-function word divide) or deep (*e.g.* long-distance information in n-gram modelling or the incorporation of parser outputs) or both, as in combined feature sets; and they are often supplemented by text-based features with varying degrees of domain-independence (*e.g.* sentence length and use of punctuation).

However, this thesis highlights a deficiency of *a priori* knowledge of prosody in both rule-based and data-driven phrase break models. Moreover, some models which do incorporate prosodic features are insufficiently linguistically-motivated: syllable counts are not best suited to a stress-timed language like English; and similarly, since nouns are highly correlated with boundaries and since primary stress in English nouns of more than one syllable tends to fall early in the word, word-final syllables minus canonical stress labels (*cf.* 3.7) are unlikely to emerge as good *categorical* boundary predictors.

There is a recent trend towards leveraging real-world knowledge to enhance performance in machine learning. The author concurs with studies that recognise how, even in silent reading, humans project prosody onto text and treat it as part of the input. Hence we have developed ProPOSEL, a tool for automatically projecting *a priori* knowledge of prosody from the lexicon onto text. This tool is also domain-independent insofar as it is compatible with English corpora tagged with four different PoS-tagging schemes.

There is consensus in the ASR community that pauses affect vowel durations in adjacent words. Based on observations from poetry and prose of an apparent correlation between a subset of English vowels and prosodic boundaries, the thesis redefines this causal relationship and interprets complex vowels (the subset) as phrase break *signifiers*. This correlation is first verified via significance testing on samples of 17th. Century English verse and contemporary British English speech. A range of dictionary-derived, symbolic prosodic features (including complex vowels) are then expressed as attribute-value sets in machine-learning experiments to explore their predictive potential in comparison with traditional phrase break features: punctuation and syntax.

10.7. Thesis main hypothesis: significance testing

Language use in poets who are native English speakers only differs in degree from that of normal English speakers:

> ‘...Since our concern was speech, and speech impelled us
> To purify the dialect of the tribe...’ (T.S.Eliot, *Little Gidding*, Part II, 1942)

A creative insight of this PhD is that if poets favour certain sounds in words as phrase break devices, then this may be endemic to the language in question (*i.e.* English) and *not* a poetic conceit.

Motivated by the observed presence of diphthongs and triphthongs at rhythmic junctures in verse, Chapter 7 of this thesis verifies a statistically significant correlation between complex vowels and phrase breaks in a representative corpus of 17th. Century English verse. Furthermore, another important finding from this study is that phrasing *variants* of the same text give the same highly correlated result.

Significance tests in Chapter 8 demonstrate how the statistically significant association between pre-boundary lexical items bearing complex vowels and gold standard phrase break annotations is upheld for contemporary British English speech in the form of a lecture (8.4; 8.5). Subsequently, a more comprehensive study of this phenomenon is then undertaken in response to recommendations from Wichman (2009), namely, to try a different genre, and one that more closely resembles spontaneous, rather than read speech: the informal news commentary in Section A of Aix-MARSEC, which comprises multiple speakers and different genders. The same statistically significant association between complex vowels and phrase breaks is again upheld in thesis sections (8.7; 8.8; 8.9) which also introduce the latest version of the algorithm used to merge the SEC and Aix-MARSEC datasets and map between the LOB and C5 tagsets.

We therefore have empirical evidence from three very different styles of speech (seventeenth century verse, a scripted lecture on economics, and informal news commentary) of a significant correlation between complex vowels and phrase breaks in English. Each dataset is relatively small, but the fact that this correlation is common to all suggests that this is a generic habit of English speech.

10.8. Thesis main hypothesis: machine learning experiments

Chapters 7 and 8 of this thesis establish a statistically significant correlation between words bearing one of a subset of the English vowel system (the subset being complex vowels) and boundary annotations which constitute intelligible and naturalistic human phrasing in different spoken genres. Given this correlation, and given the dearth of prosodic features in current phrasing models for TTS, we have evaluated the efficacy of complex vowels and other symbolic prosodic features for the phrase break prediction task, and from the evidence so far, we have concluded that at best, introducing such features does not result in significant performance gains.

Augmenting traditional feature sets of syntax and punctuation with *four novel* symbolic/descriptive prosodic features, namely complex vowels, beats, word-internal syllable-stress weightings, and word-internal rhythmic profile, *does* improve significantly on majority class baseline performance, and also improves on the baseline set by punctuation as top-performing feature – *though not significantly*. Using syntax plus the complete set of prosodic features again improves significantly on baseline performance, namely OneR prediction via syntactic identity of the word immediately following the index to be classified, but this model cannot improve on the success rate of a model using the full set of syntactic features (as opposed to a single syntactic feature) *implemented via the same J48 classifier*. Next, using syntax and complex vowels *only*, we again improve significantly on OneR baseline performance, but again cannot improve on the syntax-only model, nor can we achieve such improvements via single implementations of other prosodic features. Finally, *selective* use of prosodic information (*i.e.* complex vowels plus word-internal syllable weights and rhythmic properties in certain index positions) to complement syntax *does* lead to a better success rate than a syntax-only model but this achievement is not found to be significant. However, a final point to make is that this thesis posits not just one, but *four* new phrase break features, and breaks new ground in addressing '...the case for a symbolic, intermediate representation of prosody...’ (Ostendorf, 2009) by introducing *descriptive* (symbolic) prosodic features for phrase break prediction.

10.9. Thesis main hypothesis: conclusions

The evidence presented in this thesis tends to disprove the hypothesis that symbolic prosodic features, in addition to syntax and punctuation, will enhance overall performance in phrase break prediction for English (*cf.* Run 33). Moreover, in the absence of punctuation, the addition of such features, whether en masse or singly or selectively, has not improved on accuracy attained by a syntax-only model. It is therefore fair to say from the evidence that *at best*, introducing prosodic features does not lead to *significant* gains, and that *at worst*, prosody introduces unnecessary noise. However, these findings are based solely on a limited dataset of BBC radio transcripts – similar in size and from the same source as test sets used in other studies (*cf.* Taylor and Black, 1998; Busser *et al.*, 2001) – where the dominance of syntax as boundary predictor is to be expected; it may not apply in the case of less grammatically well-formed texts. Therefore, we recommend further testing with a larger, more varied dataset which includes, for example, spontaneous conversational speech, and/or surreptitious recordings, and/or less “expert-crafted” text such as verbal autopsy reports (Danso *et al.*, 2011). As yet, there is no such marked up corpus available for machine learning experiments so it would need to be assembled.

10.9.1. Complex vowels as phrase break signifiers

The fact that adding complex vowels as a feature does not appear to improve success rates in our phrasing model does not, we posit, detract from findings in Chapters 7 and 8 of this thesis, where complex vowels are found to be highly correlated with boundaries in a variety of genres – a contemporary British English lecture; 17th. Century English verse; informal BBC radio news commentary – and for multiple speakers. This thesis contends that native English speakers may use certain sound patterns as *linguistic signs* for phrase breaks, and that one such sign is the subset of complex vowels. We consider complex vowels as boundary precursors, as visual/textual, plus vocal and aural cues signifying optimal parsing and phrasing strategies for readers, speakers and listeners alike. We also believe this finding may apply to other languages: the preference for complex vowels in English may translate to a different subset of favoured vowel sounds in other languages. There is therefore considerable potential for further work in this area. The studies involving complex vowels in Chapters 7 and 8 uncover just *one* element in the *prosodic* and *graphemic* tiers (as distinct from the *syntactic* tier) of boundary phenomena. Moreover, as part of our recommendation for further testing, symbolic prosodic features like complex vowels may achieve better results for phrasing models in languages where word order is less constrained (*e.g.* Arabic), and where, presumably, the predictive potential of syntax may be less reliable.

10.9.2. Grey areas: data skew, different classifiers, different metrics

Any phrase break prediction experiment will encounter the problem of skewed data. Non-breaks (the majority class) will always significantly outnumber breaks (the minority class) in real world corpus data. Hence the baseline set by the majority class will always be challenging (*e.g.* 79%) for the language model and it is therefore important to validate apparent gains in accuracy via significance testing. It is also important to consider more than one evaluation metric; this is highlighted by discrepancies in performance over different classifiers implementing the same feature set. Bayesian classifiers (*e.g.* Runs 6 and 7) re-capture many more minority class instances – a priority, one might argue. This should be set against their proclivity for generating more false positives to get a true picture of performance. Experiments in Chapter 9 consider results in terms of both accuracy and balanced classification rate; and in the case of conflicting results (*e.g.* Run 6), more confident conclusions have been drawn via appropriate significance testing (§9.13.2). We therefore recommend both strategies – use of additional metrics to complement accuracy, and significance testing – for future evaluation of phrasing models.

10.9.3. Further insights gained

We have demonstrated how linguistic data arrays in ProPOSEC can be reconceptualised as training instances for machine learning. Data conversion into succinct attribute-value sets for use in WEKA is non-trivial and is implemented via a knowledge engineering algorithm which reduces the number of values for POS and lexical stress patterns, and condenses the beat, rhythm, complex vowels and punctuation attributes even further into a finite set of two or three values. Succinctness does not undermine subtlety, however: the algorithm generates a specific identity for function words whose prosodic characteristics resemble those of content words, for example. We believe that standard categorisation of function words for phrase break prediction needs revisiting: (i) pronouns are not a homogeneous group; (ii) words exhibit prosodic behaviour which belies their syntax; (iii) sparing use of fine syntactic distinctions (*e.g. that* and *of* versus other conjunctions and prepositions) pays off (*cf.* ADTree models in Runs 5 and 13a, Appendix 3).

10.10. Further work: extension and application of ProPOSEL for VTTS

A potential follow-on project from this PhD is a visual extension of the ProPOSEL lexicon, mapping already-present phonetic transcriptions to their equivalent viseme sequences for visual text-to-speech (VTTS) applications. In standard TTS, the output from the NLP system module is a text file which represents a re-working of the orthographical form into its equivalent phonetic and prosodic transcription. In VTTS, there is an additional transformation where phonetic transcriptions are mapped to sequences of visemes denoting corresponding lip shapes for production of each segment or phoneme. Phonemes are *visually* ambiguous and therefore the mapping from phonemes to visemes is many-to-one. Moreover, there is widespread discrepancy in phoneme-viseme mappings used in prototype VTTS applications for English (*cf.* Ezzat and Poggio, 2000; Kalberer & Gool, 2001; Bozkurt *et al.*, 2007). Construing ProPOSEL as a speech-to-viseme generator and VTTS component would include extending the lexicon as follows:

1. Rationalisation of a default phoneme-viseme mapping for British and/or American English.
2. Generating canonical viseme sequences from phonetic transcriptions for each lexicon entry.
3. Deriving default normalised duration vectors from a reliable speech corpus such as Aix-MARSEC (where several time-stamped prosodic annotation tiers could be explored/used) and supplementing lexicon entries with this information.

10.11. Applying thesis Text Analytics techniques to Arabic

Another follow-on project is to apply thesis Text Analytics techniques for *English* (§Chapters 7 and 8) to explore phrasing strategies in another stress-timed language, *Arabic*. Here, an initial dataset would be the Quran, since certain (*Tajweed*) editions of this text already incorporate fine-grained prosodic boundary annotations, and a morphologically and syntactically annotated Quran corpus is also available (Dukes, 2011). The project would involve: (i) mining these prosodic-syntactic boundaries for syntactic, phonetic and prosodic correlates, and verifying frequent patterns via significance testing; and (ii) re-expressing significant sound patterns as symbolic features to be incorporated and tested in state-of-the-art phrase break models for both Classical and Modern Standard Arabic TTS. Interestingly, it is *less* likely that punctuation would be available as a feature in such applications; and also, we might speculate that since word order in Arabic is less restricted than English, syntax may be less reliable as a boundary predictor, and thus symbolic prosodic features might indeed bring tangible improvements. A further idea would be to use *a priori* knowledge of the sound system of Arabic inherent in Quranic boundary annotations to inform phoneme-viseme mappings for Arabic VTTS applications.

10.12. Summary: PhD impact, originality, and contribution to research field

This final section presents a brief summary of research contributions and achievements of this PhD.

10.12.1. Understanding the prosody-syntax interface: discursive analysis and experimentation

1. An important distinction is made in the Introduction between *prosodic chunking and highlighting*, and these concepts are integrated into later discussions of prosodic variance (*cf.* 5.3.4).
2. There is sustained, discursive analysis of *gold standard phrase break annotations* in both SEC (the Spoken English Corpus) and Milton's '*Paradise Lost*,' plus presentation and discussion of *variant phrasing* in same. Similar treatment is given to *prepositional phrase attachment* and the crucial

distinction between particles and prepositions for accurate and naturalistic automated chunking.

3. The author believes her most original insight and research contribution is: (i) the observation; and (ii) the discovery via significance testing, of co-occurrence between words bearing complex vowels and phrase breaks first in poetry, and then in different spoken genres.
4. The above experimental work relies on two further important thesis insights, namely: (i) the lack of *a priori* knowledge of prosody to complement punctuation and syntax in language models for predicting prosodic phenomena (*i.e.* phrase breaks); and (ii) that prosody can be projected onto text in much the same way as syntax is.

10.12.2. Artefact and resource creation

1. Python and NLTK have been used to build ProPOSEL, a customised prosody and part-of-speech English lexicon for text annotation and Text Analytics.
2. ProPOSEL is supported with software tools and an extended user tutorial.
3. ProPOSEL is then used to create the ProPOSEC dataset, supplementing annotations in SEC and Aix-MARSEC with canonical annotations from the lexicon. There is then further demonstration of how ProPOSEC annotations can be converted into attribute-value sets for machine learning in WEKA.

10.12.3. Originality and ideas for further work

1. This thesis supplements traditional phrase break features (punctuation and syntax) with categorical features derived from the lexicon. This constitutes an original way of representing prosody, as opposed to continuous features such as fundamental frequency. Four such novel symbolic features are posited and evaluated.
2. Another emerging hypothesis is that: (i) other correlations may emerge between English phonemes and phrase breaks; and (ii) the principle of annotating text with a ProPOSEL-like tool and then mining these annotations for boundary correlates can be applied to other languages *e.g.* Arabic.
3. An interesting application of ProPOSEL might be integration into dialogue systems for linguistically-informed lip-synch in relational agents.

10.12.4. Impact

1. Journal and conference publications arising from this thesis have addressed a range of research communities: Corpus Linguistics; Natural Language Processing; Linguistic Resources and Evaluation; the speech community (*i.e.* Interspeech and Speech Prosody); and Literary and Linguistic Computing/Digital Humanities.

References

Abney S. 1991. ‘Parsing by Chunks.’ In *Principle-Based Parsing: Computation and Psycholinguistics*. Berwick, R.C., Abney S. and Tenny, C. (eds.). Dordrecht. Kluwer Academic Publishers.

Abney S. 1992. ‘Prosodic Structure, Performance Structure and Phrase Structure.’ *Proceedings, Speech and Natural Language Workshop 1992*: 425-428.

Abney, S. 1994. ‘Partial Parsing.’ Tutorial. *Proceedings, ACL Conference on Applied Natural Language Processing, ANLP'94*. Accessed January 2010: http://vinartus.net/spa/94j.pdf

Abney S. 1995. ‘Chunks and Dependencies: Bringing Processing Evidence to Bear on Syntax.’ *Computational Linguistics and the Foundations of Linguistic Theory*. CSLI. (1995).

Ananthakrishnan, S. and Narayanan, S.S. 2008. ‘Automatic Prosodic Event Detection Using Acoustic, Lexical, and Syntactic Evidence.’ *IEEE Transactions on Audio, Speech, and Language Processing, TASLP 2008*. Vol. 16.1: 216-228.

Atterer M. 2002. ‘Assigning Prosodic Structure for Speech Synthesis: a Rule-Based Approach.’ *Proceedings, 1st. International Conference in Speech Prosody, SP’02.*

Atterer M. and Klein E. 2002. ‘Integrating Linguistic and Performance-Based Constraints for Assigning Phrase Breaks.’ *Proceedings, Coling 2002*. 29-35.

Atwell, E., Hughes, J. and Souter, C. 1994. ‘AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models.’ In Klavens, J. and Resnik, P. (eds.) *The Balancing Act - Combining Symbolic and Statistical Approaches to Language*. Proceedings of the Workshop in conjunction with the 32nd. Annual Meeting of the Association of Computational Linguistics.

Atwell, E. 1996. ‘Comparative evaluation of grammatical annotation models.’ In Sutcliffe, R., Koch, H. and McElligott, A. (eds.) *Industrial Parsing of Software Manuals*. 25-46. Rodopi.

Atwell, E., Demetriou, G., Hughes, J., Schriffin, A., Souter, C. and Wilcock, S. 2000. ‘A comparative evaluation of modern English corpus grammatical annotation schemes.’ *ICAME Journal*. Vol. 24: 7-23.

Atwell, E. 2008. 'Development of tag sets for part-of-speech tagging.' In Ludeling, A. and Kyto, M. (eds.) *Corpus Linguistics: An International Handbook*. Mouton de Gruyter.

Auran, C., Bouzon, C. and Hirst, D. 2004. 'The Aix-MARSEC Project: An Evolutive Database of Spoken English.' *Proceedings, Speech Prosody 2004*. 561-564. Accessed: January 2010: http://www.isca-speech.org/archive/sp2004/sp04_561.pdf

Baayen, R. H., Piepenbrock, R. and Gulikers, L. 1996. *CELEX-2*. Linguistic Data Consortium. Philadelphia.

Baldwin, T., Bender, E., Flickinger, D., Kim, A., and Oepen, S. 2004. 'Roadtesting the English Resource Grammar over the British National Corpus.' *Proceedings, LREC-2004.*

Banks, Jr., T.H. 1927. 'A Study of the Relation of the Full Stops to the Rhythm of Paradise Lost.' *Proceedings of the Modern Languages Association*. 42.1: 140-5

Barber, C. 1997. *Early Modern English.* Edinburgh. Edinburgh University Press

Bayerl P. and Paul, K. 2007. 'Identifying sources of disagreement: Generalizability theory in manual annotation studies.' *Computational Linguistics*, Vol. 33.1: 3-8.

BBC. 2010. "The Great Vowel Shift." (Creative content for online encyclopedia from h2g2 researcher community). Accessed: January 2010. http://www.bbc.co.uk/dna/h2g2/classic/A964578

Beckman, M.E. and Ayers, G.M. 1997. Guidelines for ToBI Labelling. Department of Linguistics, Ohio State University Accessed: January 2010. http://www.ling.ohio-state.edu/research/phonetics/E_ToBI/ToBI/ToBI.1.html

Bell, A., Brenier, J.M., Gregory, M., Girand, C. and Jurafsky, D. 2009. 'Predictability effects on durations of content and function words in conversational English.' *Memory and Language*, Vol. 60: 92-111.

Bell, P. 2005. *Adaptation of Prosodic Phrasing Models*. MPhil Thesis. University of Cambridge. Accessed January 2010. http://homepages.inf.ed.ac.uk/s0566164/m_thesis.pdf

Bird, S., Klein, E. and Loper, E. 2009. *Natural Language Processing with Python.* Sebastopol, CA. O'Reilly Media, Inc.

Black A.W., Taylor, P. and Caley R. 1999. 'The Festival Speech Synthesis System: System Documentation.' Festival Version 1.4. Accessed January, 2010: http://www.cstr.ed.ac.uk/projects/festival/manual/festival_17.html

Black A.W., Taylor, P. and Caley R. 2000. 'Speech Synthesis in Festival: A Practical Course in Making Computers Talk.' Festival Version 2.0. Accessed January, 2010: http://festvox.org/festtut/notes/festtut_toc.html

Boersma, P. and Weenink, D. 2009. 'Praat: doing phonetics by computer (version 5.1.05).' [Computer program]. Accessed: January 2010. http://www.praat.org/

Bouzon, C. and Hirst, D. 2004. 'Isochrony and Prosodic Structure in British English.' *Proceedings, Speech Prosody 2004.* 223-226.

Bozkurt, E., Erdem, C.E., Erzin, E., Erdem, T. and Ozkan, M. 2007. 'Comparison of Phoneme and Viseme Based Acoustic Units for Speech Driven Realistic Lip Animation.' In *Signal Processing and Communications Applications*, 2007. SIU 2007. IEEE 15th. pp. 1-4.

Bridges, R. 1921. *Milton's Prosody: with a Chapter on Accentual Verse and Notes by Robert Bridges*. Oxford. Oxford University Press

Brierley, C. and Atwell, E. 2009. 'Exploring Imagery in Literary Corpora with the Natural Language ToolKit.' *In Proc. Corpus Linguistics 2009*

Burnage, G. 1990. *CELEX - A Guide for Users.* Nijmegen: Centre for Lexical Information. University of Nijmegen. Accessed: 2009. http://www.ru.nl/celex/subsecs/section_doc.html

Burnard, L. (ed.) 2000. *Reference Guide for the British National Corpus (World Edition).* Accessed January 2010. http://www.natcorp.ox.ac.uk/docs/userManual/

Busser B., Daelemans W. and van den Bosch A. 2001. 'Predicting phrase breaks with memory-based learning.' *Proceedings, 4th. ISCA Tutorial and Research Workshop on Speech Synthesis, 2001.*

Carletta, J. 1996. 'Assessing agreement on classification tasks: the kappa statistic.' *Computational Linguistics.* Vol. 22.2: 249-254.

Carnegie-Mellon Univeristy. 1998. *The CMU Pronouncing Dictionary* (Version 0.6). Accessed January 2010: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Chandler, D. 2002. *Semiotics: the Basics.* London. Routledge

Croft, W. 1995. ‘Intonation Units and Grammatical Structure.’ *Linguistics*. 33: 839-882

Danso, S., Atwell, E., Johnson, O., Asbroek, G., Soromekun, S., Edmond, K., Hurt, C., Hurt, L., Zandoh, C., Tawiah, C., Hill, Z., Fenty, J., Amenga Etego, S., Owusu Agyei, S. and Betty R Kirkwood. 2011. ‘Verbal autopsy corpus for machine learning cause of death.’ *Proceedings, Corpus Linguistics 2011*. Birmingham, UK.

Davel, M. and Barnard, E. 2008. ‘Pronunciation Prediction with Default&Refine.’ *Computer Speech and Language*. 22.4: 374-393.

Dehé, N. and Wichmann, A. 2010. ‘Sentence-initial I think (that) and I believe (that): Prosodic evidence for uses as a main clause, comment clause and discourse marker.’ In *Studies in Language* 34.1. John Benjamins.

Dukes, K. 2011. *The Quranic Arabic Corpus*. Online. Accessed: August 2011. http://corpus.quran.com

Durusau, P., O'Donnell, M. 2002. ‘Concurrent Markup for XML Documents.’ Presentation at XML Europe 2002. Accessed: February 2009. http://www.idealliance.org/papers/xmle02/dx_xmle02/papers/03-03-07/03-03-07.html

Elliott, John. 2003. ‘Unsupervised learning of word classes using function word constraints.’ *Proceedings, International Conference on Computational Ntaural Language Learning*, 25-30. ACL; Daelemans, W. (editors).

Ezzat, T. and Poggio, T. 2000. ‘Visual Speech Synthesis by Morphing Visemes.’ In *International Journal of Computer Vision*. Vol. 38.1. pp. 45-57

Fodor, J. D. 2002. ‘Psycholinguistics Cannot Escape Prosody.’ *Proceedings, Speech Prosody (SP-2002)*. 83-90.

Furui, S. 2009. ‘Selected topics from 40 years of research on speech and speaker recognition.’ Keynote Speech. *Interspeech 2009.*

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L. and Zue, V. 1993. *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium, Philadelphia

Gazzah, S. and Ben Amara, N.E. 2008. ‘New Over-Sampling Approaches Based on Polynomial Fitting for Imbalanced DataSets.’ *Proceedings 8th IAPR Workshop on Document Analysis Systems*, 677-84.

Gee, J. P. and Grosjean, F. 1983. ‘Performance Structures: A Psycholinguistic and Linguistic Appraisal.’ *Cognitive Psychology*. Volume 15: 411-458.

Geraghty, J. 2003. *Digital Facsimile Project.* Accessed: Janaury 2010. http://www.johngeraghty.com/Literature/Texts/Milton/P_Lost_1674/PL_6.jpg

Godfrey, J.J. and Holliman, E. 1997. *Switchboard-1 Release 2*. Linguistic Data Consortium, Philadelphia.

Grabe, E. 2001. ‘Prosodic Annotation.’ PowerPoint. *9th ELSNET European Summer School on Language and Speech Communication*, *Prague*. Accessed: 2006.

Grabe, E., Post, B. and Nolan, F. 2001. ‘Modelling intonational Variation in English. The IViE system.’ In Puppel, S. and Demenko, G. (eds). *Proceedings of Prosody 2000*: 51-57. Accessed: January 2010. http://www.phon.ox.ac.uk/files/apps/old_IViE/#pro

Grabe, E., Kochanski, G. and Coleman, J. 2003. ‘Quantitative modelling of intonational Variation.’ *Proceedings, Speech Analysis and Recognition in Technology, Linguistics and Medicine 2003*. Accessed: January 2010. http://www.phon.ox.ac.uk/files/apps/oxigen/

Greenbaum, S. and Svartvik, J. 1990. *The London Corpus of Spoken English: Description and Research.* Lund University Press. Accessed: January 2010. http://khnt.hit.uib.no/icame/manuals/londlund/index.htm

Grosjean, F., Grosjean, L. and Lane, H. 1979. ‘The Patterns of Silence: Performance Structures in Sentence Production.’ *Cognitive Psychology*. Volume 11: 58-81.

Gussenhoven, C., Rietveld, T., Kerkhoff, J. and Terken, J. 2003. ‘ToDI second edition (2003).’ Accessed: January 2010. http://todi.let.kun.nl/ToDI/home.htm

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. 2009. The WEKA Data Mining Software: An Update. In *SIGKDD Explorations*. 11: 1.

Hall, P. 2004. *Shakespeare’s Advice to the Players.* London. Oberon Books Ltd.

Hartikainen, E., Maltese, G., Moreno, A., Shammass, S. and Ziegenhain, U. 2003. ‘Large Lexica for Speech-to-Speech Translation: From Specification to Creation.’ *Proceedings, EUROSPEECH-2003*: 1529-1532.

Hazen, T.J., Hetherington, I.L., Shu, H. and Livescu, K. 2005. ‘Pronunciation Modelling using a Finite State Transducer Representation.’ *Speech Communication.* 46.2: 189-203.

Helleputte, T. and Dupont, P. 2009. 'Partitally supervised feature selection with regularized linear models.' *Proceedings of 26th International Conference on Machine Learning*. ACM, New York.

Higgins, J. 2010. *Homographs*. Accessed: January 2010. http://myweb.tiscali.co.uk/wordscape/wordlist/homogrph.html

Hirschberg J. 2002. 'Communication and Prosody: The Functional Aspects of Prosody.' *Speech Communication*. 36.1: 31-43.

Hirschberg J. and Prieto P. 1996. 'Training Intonational Phrasing Rules Automatically for English and Spanish Text-to-speech.' *Speech Communication*. 18.3: 281-290.

Hirst, D. 2009. 'The rhythm of text and the rhythm of utterances: from metrics to models.' Proceedings, *INTERSPEECH-2009*. 1519-1522.

Hirst, D., Di Cristo, A. and Espesser, R. 2000. 'Levels of representation and levels of analysis for the description of intonation systems.' In Horne, M. (ed.) 2000 *Intonation: Theory and Experiment*. Dordrecht. Kluwer Academic Press. Accessed: January 2010. http://aune.lpl.univ-aix.fr/~hirst/publis.html#recent

Holdcroft, D. 1991. *Saussure: Signs, System, and Arbitrariness*. Cambridge. Cambridge University Press

Hornby, A. S. 1974. *Oxford Advanced Learner's Dictionary of Current English*. (Third edition). Oxford. Oxford University Press.

Huckvale, M. 2002. 'Speech Synthesis, Speech Simulation and Speech Science.' *Proceedings, International Conference on Speech and Language Processing* '02. 1261-1264.

Ingulfsen, T., Burrows, T. and Buchholz, S. 2005. 'Influence of Syntax on Prosodic Boundary Prediction.' *Proceedings, INTERSPEECH 2005*. 1817-1820.

Jackson, M.P. 2002. 'Pause Patterns in Shakespeare's Verse: Canon and Chronology.' *Literary and Linguistic Computing*. 17.1. 37-46.

Jenkins, H. (ed.) 2003. *The Arden Shakespeare: Hamlet*. London. Thomson Learning.

Johansson, S., E. Atwell, R. Garside and G. Leech. 1986. *'The Tagged LOB Corpus: Users' manual.'* Bergen. Norwegian Computing Centre for the Humanities. Accessed: January. 2010. http://khnt.hit.uib.no/icame/manuals/lobman/INDEX.HTM%20

Jurafsky, D. and Martin, J.H. 2000. *Speech and Language Processing*. New Jersey. Prentice-Hall Inc.

Jurafsky, D. and Martin, J.H. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Draft of 2nd. edition). Accessed: 2009. http://www.cs.colorado.edu/~martin/slp2.html

Kalberer, G.A. and Gool, L.V. 2001. 'Face Animation Based on 3D Speech Dynamcis.' In *Proceedings of 14th. Conference on Computer Animation, 2001*. 20-27.

Kempen, G. and Hoenkamp, E. 1982. 'Incremental Sentence Generation: Implications for the Structure of a Syntactic Parser.' Proceedings, 9*th*. *International Conference in Computational Linguistics (Coling '82).*

Kessler, B. and Treiman, R. 2001. 'Relationships between Sounds and Letters in English Monosyllables.' *Journal of Memory and Language.* 44.4: 592-617.

Kingsbury, P., Strassel, S., McLemore, C. and MacIntyre, R. 1997. *CALLHOME American English Lexicon (PRONLEX).* Linguistic Data Consortium: Philadelphia.

Koehn P., Abney S., Hirschberg J. and Collins, M. 2000. 'Improving Intonational Phrasing with Syntactic Information.' *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000. Vol 3: 1289-1290.

Knill, D. 2009. Personal communication. Interspeech 2009.

Knowles, G. 1987. *Patterns of Spoken English: An Introduction to English Phonetics.* Harlow. Longman Group UK Ltd.

Knowles, G. 1996a. 'From text structure to prosodic structure.' In *Working with Speech: Perspectives on research into the Lancaster/IBM Spoken English Corpus*. Knowles, G., Wichman, A. and Alderson, P. (eds.). Harlow. Addison Wesley Longman Ltd.

Knowles, G. 1996b. 'The value of prosodic transcriptions.' In *Working with Speech: Perspectives on research into the Lancaster/IBM Spoken English Corpus*. Knowles, G., Wichman, A. and Alderson, P. (eds.). Harlow. Addison Wesley Longman Ltd.

Kurimo M., Creutz M., Varjokallio M., Arisoy E. and Saraclar M. 2006. 'Unsupervised segmentation of words into morphemes - Challenge 2005: An Introduction and Evaluation Report.' In: Kurimo, M., Creutz, M. and Lagus, K. (eds.) *Proceedings of the PASCAL Challenge '06 Workshop on Unsupervised Segmentation of Words into Morphemes*.

Ladd, R. 1996. *Intonational Phonology* Cambridge. Cambridge University Press.

Langworthy, C.A. 1931. 'A Verse-Sentence Analysis of Shakespeare's Plays.' *Proceedings of the Modern Languages Association*. 46.3: 738-51.

Leavis, Q.D. 1965. *Fiction and the Reading Public*. London. Chatto and Windus.

Leech, G. and Smith, N. 2000. 'Manual to accompany The British National Corpus (Version 2) with Improved Word-class Tagging.' Accessed: January 2010. http://www.natcorp.ox.ac.uk/docs/bnc2postag_manual.htm

Lessard, G. and Levison, M. 2005. 'Computational Generation of Limericks.' *Literary and Linguistic Computing*. 20: 89-105.

Liberman, M.Y. and Church, K.W. 1992. 'Text Analysis and Word Pronunciation in Text-to-Speech Synthesis.' In *Advances in Speech Signal Processing*. Furui S. and Sondhi, M.M. (eds.). New York. Marcel Dekker Inc.

Luxon, T.H. (ed.) 2010. 'The Milton Reading Room.' Accessed: Janauray 2010. http://www.dartmouth.edu/~milton

McCarthy, M. 1991. *Discourse Analysis for Language Teachers.* Cambridge. Cambridge University Press.

Maidment, J. 2009. 'The Speech Internet Dictionary.' Accessed: January 2010. http://www.phon.ucl.ac.uk/home/johnm/sid/sidhome.htm

Manning, C.D. and Schutze, H. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts. The Massachusetts Institute of Technology.

Marcus, M.P., Santorini, B. And Marcinkiewicz, M.A. 1994. 'Building a Large Annotated Corpus of English: The Penn Treebank.' *Computational Linguistics.* 19.2: 313-330.

Maskey, S.R., Black, A.W. and Tomokiyo, L.M. 2004. 'Bootstrapping Phonetic Lexicons for New Languages.' *Proceedings, INTERSPEECH-2004*. 69-72.

Menzer, M.J. 2000. 'The Great Vowel Shift.' Accessed: January 2010. http://facweb.furman.edu/~mmenzer/gvs/index.htm

Merlo P. and Ferrer E.E. 2006. 'The Notion of Argument in Prepositional Phrase Attachment.' *Computational Linguistics.* 32.3: 341-378.

Miller, G.A. 1956. 'The magical number seven, plus or minus two: Some limits on our capacity for processing information.' *The Psychological Review*. 63: 81-97.

Mitton, R. 1992. *A description of a computer-usable dictionary file based on the Oxford Advanced Learner's Dictionary of Current English*. Accessed: 2009. http://comp.lin.msu.edu/stabler-notes/1850/ascii_0710-2.txt

Mitton, R. 1996. *English Spelling and the Computer*. London. Longman.

Mortimer, C. 1985. *Elements of Pronunciation*. Cambridge. Cambridge University Press.

Naber, D. 2003. *A Rule-based Style and Grammar Checker*. Accessed: January 2010. http://www.danielnaber.de/languagetool/download/style_and_grammar_checker.pdf

Nancarrow, O. and Atwell, E. 2007. 'A Comparative Study of the Tagging of Adverbs in Modern English Corpora.' *Proceedings of Corpus Linguistics 2007*.

Oras, A. 1960. *Pause Patterns in Elizabethan and Jacobean Drama*. Gainesville, FL. University of Florida Press.

Ostendorf, M. 2010. 'Representations of Prosody in Computational Models for Language Processing.' Keynote Lecture. *SPEECH PROSODY 2010*.

Ostendorf, M., Price, P. and Shattuck-Hufnagel, S. 1996. *Boston University Radio Speech Corpus*. Linguistic Data Consortium, Philadelphia

PASCAL Thematic Programme 2008. Accessed: January 2010. http://www.cs.man.ac.uk/~neill/thematic08.html

Paulin, T. 2003. 'Spirit of the Age.' *The Guardian*. Saturday 5 April, 2003.

Pedler, J. 2001. Computer spellcheckers [*sic*] and dyslexics—a performance survey. *British Journal of Educational Technology*, 32.1 23–37.

Pedler, J. 2007. *Computer Correction of Real-word Spelling Errors in Dyslexic Text*. PhD Thesis. Birkbeck: London University.

Pedler, J. and Mitton, R. 2003. *CUVPlus*. Arts and Humanities Data Service. Accessed: January 2010. http://ahds.ac.uk/catalogue/collection.htm?uri=lll-2469-1

Peppe, S. 2006. Personal Communication. University of Leeds.

Pickering, B., Williams, B. and Knowles, G. 1996. 'Analysis of transcriber differences in SEC.' In *Working with Speech: Perspectives on research into the Lancaster/IBM Spoken English Corpus*. Knowles, G., Wichman, A. and Alderson, P. (eds.). Harlow. Addison Wesley Longman Ltd.

Pitrelli, J., Beckmann, M. and Hirschberg, J. 1994. 'ToBI (Tones and Break Indices).' *Proceedings, 1994 International Conference on Spoken Language Processing*: 18-22.

Ramshaw, L. A. and Marcus, M. P. 1995. 'Text chunking using transformation-based learning.' *Proceedings, 3rd. ACL Workshop on Very Large Corpora*. 82-94.

Rayson, P., Archer, D. and Smith, N. 2005. 'VARD versus Word: A comparison of the UCREL variant detector and modern spell checkers on English historical corpora.' *Proceedings of Corpus Lingusitics Conference Series*. Vol.1.1.

Rayson, P., Archer, D., Baron, A., Culpeper, J. and Smith, N. 2007. 'Tagging the Bard: Evaluating the accuracy of a modern POS tagger on Early Modern English corpora.' Proceedings, *Corpus Linguistics*.

Read I. and Cox S. 2004. 'Using part-of-speech for predicting phrase breaks.' *Proceedings, INTERSPEECH-2004*. 741-744.

Read, I. and Cox, S. 2005. 'Stochastic and syntactic techniques for predicting phrase breaks.' *Proceedings, INTERSPEECH-2005*. 3233-3236.

Read, I. and Cox, S. 2007. 'Stochastic and Syntactic Techniques for Predicting Phrase Breaks.' *Computer Speech and Language*. Vol. 21.3: 519-542.

Roach, P. 2000. *English Phonetics and Phonology: A Practical Course* (3rd. edition). Cambridge. Cambridge University Press.

Roach, P., Knowles, G., Varadi, T. and Arnfield, S. 1993. 'Marsec: A machine-readable spoken English corpus.' *Journal of the International Phonetic Association*. Vol. 23.1: 47-53.

Rogers, W.E. 2000. 'The History of English Phonemes.' Accessed: January 2010. http://facweb.furman.edu/~wrogers/phonemes/

Santorini, B. 1990. *Part-of-Speech tagging guidelines for the Penn Treebank Project.* Technical report: MS-CIS-90-47. University of Pennsylvania.

Schmid, H. and Atterer, M. 2004. 'New Statistical Methods for Phrase Break Prediction.' *Proceedings, 20th. International Conference on Computational Linguistics, Coling 2004*. 659-665.

SETI. 2011. *DRAKE EQUATION*. Accessed: August 2011. SETI Institute. http://www.seti.org/drakeequation

Shriberg, L. and Lof, G. 1991. ‘Reliability studies in broad and narrow phonetic transcription.’ *Clinical Linguistics and Phonetics.* 5.3: 225-279.

Sinclair, J.M., Mauranen, A. 2006. *Linear Unit Grammar: Integrating Speech and Writing.* Amsterdam. John Benjamins Publishing Company

Speer, S., Warren, A. and Schafer, P. 2000. ‘Intonational Disambiguation in Sentence Production and Comprehension.’ *Journal of Psycholinguistic Research*. Vol. 29.2: 169-182.

Stallings, L.M., MacDonald, M.C. and O’Seaghdha. 1998. ‘Phrasal Ordering Constraints in Sentence Production: Phrase Length and Verb Disposition in Heavy-NP Shift.’ *Memory and Language*, Vol. 39.3: 392-417.

Stock, O. and Strapparava. 2003. ‘HAHAcronym: Humorous Agents for Humorous Acronyms. ’ *International Journal of Humor Research*. 16.1: 297-314.

Taylor, L.J. and Knowles, G. 1988. **‘**Manual of Information to Accompany the SEC Corpus: The machine readable corpus of spoken English.’ Accessed: January 2010. http://khnt.hit.uib.no/icame/manuals/sec/INDEX.HTM

Taylor, L. 1996. ‘The correlation between punctuation and tone group boundaries.’ In *Working with Speech: Perspectives on research into the Lancaster/IBM Spoken English Corpus*. Knowles, G., Wichman, A. and Alderson, P. (eds.). Harlow. Addison Wesley Longman Ltd.

Taylor, P. and Black, A.W. 1998. ‘Assigning Phrase-Breaks from Part-of-Speech Sequences.’ In *Computer Speech and Language*. 12.2: 99-117.

Taylor, P. 2000. ‘Analysis and synthesis of intonation using the Tilt model.’ *Journal of the Acoustical Society of America*. Vol. 107.3: 1697-1714. Accessed: January 2010. http://svr-www.eng.cam.ac.uk/~pat40/

Temperley, D., Sleator, D. and Lafferty, J. 2009. *Link Grammar.* Accessed January 2010: http://www.link.cs.cmu.edu/link/

UCREL. 2010. ‘CLAWS part-of-speech tagger for English.’ University Centre for Computer Research on Language, University of Lancaster. Accessed: January 2010. http://ucrel.lancs.ac.uk/claws/

University of Bergen. 1993. *The Bergen Corpus of London Teenage Language (COLT)* Department of English, University of Bergen. Accessed: January 2010. http://www.hf.uib.no/i/Engelsk/COLT/index.html

V&A. 2010. Blank Verse. *Victoria and Albert Museum.* Accessed: January 2010. http://www.vam.ac.uk/activ_events/adult_resources/creative_writing/prose_poetry_techniques/

Valve Developer Community. 2011. FacePoser Reference. Accessed: 18.01.2011. http://developer.valvesoftware.com/wiki/Faceposer_reference

Van Zaanen, M., Roberts, A.and Atwell, E. 2004. 'A multilingual parallel parsed corpus as gold standard for grammatical inference evaluation.' In Kranias, L., Calzolari, N., Thurmair, G., Wilks, Y., Hovy, E., Magnusdottir, G., Samiotou, A. and Choukri, K. (eds.) *Proceedings of LREC'04 Workshop on The Amazing Utility of Parallel and Comparable Corpora*, 58-61. European Language Resources Association.

Vergyri, D., Stolcke, A., Gadde, V.R.R., Ferrer, L. and Shriberg, E. 2003. 'Prosodic Knowledge Sources for Automatic Speech Recognition.' *Proceedings, International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*. 208-211.

Vincent, R.D., Goldberg, R.K. and Titone, D.A. 2006. 'Anagram Software for Cognitive Research that Enables Specification of Psycholinguistic Variables.' *Behaviour Research Methods*. 38.2: 196-201.

Walkinshaw, N., Bogdanov, K., Damas, C., Lambeau, B. and Dupont, P. 2010. 'A framework for competitive evaluation of model inference techniques.' *Proceedings, 1st International Workshop on Model Inference Testing*. ACM, New York.

Wang M.Q. and Hirschberg J. 1991. 'Predicting intonational phrasing from text.' *Proceedings 29th. annual meeting, Association for Computational Linguistics ACL'91*: 285-292.

Welby, P. 2003. 'Effects of Pitch Accent Position, Type, and Status on Focus Projection.' *Language and Speech*. Vol. 46.1: 53-81.

Wichmann, A. Personal communication. ICAME 30, Lancaster. 2009.

Winograd, T. 1984. 'Computer Software for Working with Language.' *Scientific American* 251: 31-45.

Wilson-Knight, G. (2001). *The Wheel of Fire*. London: Routledge Classics

Appendix 1: Mapping for Variant Syntactic Annotation Schemes in Current Version of ProPOSEL

C5 PoS tags	**Variant Syntactic Annotation Schemes and Tags, with comments and most recent revisions given in *italics***	
AJ0	Penn	JJ
	LOB	JJ,JJB,JNP
	C7	JJ,JK
AJC	Penn	JJR
	LOB	JJR
	C7	JJR
AJS	Penn	JJS
	LOB	JJT
	C7	JJT
AT0	Penn	DT
	LOB	AT,ATI
	C7	AT,AT1
AV0	Penn	RB,RBR,RBS
	LOB	QL,QLP,RB,RI,RBR,RBT,RN *(Shall I add: ABL?)*
	C7	*BCL,RA,REX,RG,RR,RL,RGR,RGT,RRR,RRT,RT*
AVP	Penn	RP
	LOB	RP
	C7	RP,RPK
AVQ	Penn	WRB
	LOB	WRB
	C7	RGQ,RGQV,RRQ,RRQV
CJC	Penn	CC
	LOB	CC
	C7	CC,CCB
CJS	Penn	CS
	LOB	CS
	C7	CS,CSA,CSN,CSW
CJT	Penn	IN,CS
	LOB	CS
	C7	CST
CRD	Penn	CD
	LOB	CD,CD1,CD1S,CDS,CD-CD *(added: NN0, NN02 UCREL)*
	C7	*MC,MC1,MC2,MF,MCMC*
DPS	Penn	*PRP$*
	LOB	PP$
	C7	APPGE
DT0	Penn	DT,PDT
	LOB	DT,DTI,DTS,DTX,ABL,ABN,ABX,AP,APS
	C7	DB,DB2,DA,DA1,DA2,DAR,DAT,DD,DD1,DD2
DTQ	Penn	*WDT*
	LOB	WDT,WQL *(added: WDTR Shall I add WP, WP$?)*
	C7	DDQ,DDQGE,DDQV
EX0	Penn	EX

	LOB	EX
	C7	EX
ITJ	Penn	UH
	LOB	UH
	C7	UH
NN0	Penn	NN
	LOB	NNU
	C7	*NN,NNU,NNA,NNB,NNO (Remove NN0, NNA, NNB (UCREL) & add NNJ)*
NN1	Penn	NN
	LOB	NN,NNP,NR
	C7	NN1,NNT1,NNU1,ND1
NN2	Penn	NNS
	LOB	NNS,NRS,NNPS,NNUS
	C7	NN2,NNJ2,NNT2,NNU2,NNO2 *(added: NPM2 (Lancs) remove NN02)*
NP0	Penn	NNP,NNPS
	LOB	NP,NPL,NPLS,NPS,NPT,NPTS *(removed: NR, NRS)*
	C7	*NNP,NNPS*,NPD1,NPD2,NPM1,*NPM2,NNL1,NNL2,NP,NP1,NP2 (remove NPM2, NNP, NNPS & add: NNA, NNB)*
NULL	Penn	NIL - no equivalent tag
	LOB	NIL - no equivalent tag
	C7	NULL
ORD	Penn	JJ
	LOB	OD
	C7	*MD*
PNI	Penn	*NN,NP (Treebank check)*
	LOB	PN
	C7	PN,PN1
PNP	Penn	*PRP*
	LOB	PP3AS,PP3O,PP3OS,PP$$,PP1A,PP1AS,PP1O,PP1OS,PP2,PP3,PP3A
	C7	PPGE,PPIS1,PPIS2,PPIO1,PPIO2,PPY,PPH1,PPHS1,PPHS2,PPHO1,PPHO2
PNQ	Penn	*WP,WP$*
	LOB	WP,WPA,WPO,WP$ *(added: WP$R, WPOR, WPR)*
	C7	PNQV,PNQS,PNQO
PNX	Penn	PXP
	LOB	PPL,PPLS
	C7	PNX1,PPX1,PPX2
POS	Penn	POS
	LOB	$
	C7	GE
PRF	Penn	IN
	LOB	IN
	C7	IO
PRP	Penn	IN
	LOB	IN
	C7	II,IF,IW
TO0	Penn	TO
	LOB	TO
	C7	TO
UNC	Penn	FW,LS,SYM,$
	LOB	&FO,&FW,NC
	C7	FW,FU,FO

VBB	Penn	*VB,VBP*
	LOB	BEM,BER *(Removed: VB,BE,)*
	C7	VB0,VBM,VBR
VBD	Penn	VBD
	LOB	BED,BEDZ
	C7	VBDR,VBDZ
VBG	Penn	VBG
	LOB	BEG
	C7	VBG
VBI	Penn	*VB*
	LOB	BE
	C7	VBI
VBN	Penn	VBN
	LOB	BEN
	C7	VBN
VBZ	Penn	VBZ
	LOB	BEZ
	C7	VBZ
VDB	Penn	*VB,VBP*
	LOB	DO *(Removed: VB)*
	C7	VD0
VDD	Penn	VBD
	LOB	DOD
	C7	VDD
VDG	Penn	VBG
	LOB	VBG
	C7	VDG
VDI	Penn	VB
	LOB	DO
	C7	VDI
VDN	Penn	VBN
	LOB	VBN
	C7	VDN
VDZ	Penn	VBZ
	LOB	DOZ
	C7	VDZ
VHB	Penn	*VB,VBP*
	LOB	HV *(Removed: VB)*
	C7	VH0
VHD	Penn	VBD
	LOB	HVD
	C7	VHD
VHG	Penn	VBG
	LOB	HVG
	C7	VHG
VHI	Penn	VB
	LOB	HV
	C7	VHI
VHN	Penn	VBN
	LOB	HVN
	C7	VHN

VHZ	Penn	VBZ
	LOB	HVZ
	C7	VHZ
VM0	Penn	MD
	LOB	MD
	C7	VM,VMK
VVB	Penn	*VB,VBP*
	LOB	VB
	C7	VV0
VVD	Penn	VBD
	LOB	VBD
	C7	VVD
VVG	Penn	VBG
	LOB	VBG
	C7	VVG,VVGK
VVI	Penn	VB
	LOB	VB
	C7	VVI
VVN	Penn	VBN
	LOB	VBN
	C7	VVN,VVNK
VVZ	Penn	VBZ
	LOB	VBZ
	C7	VVZ
XX0	Penn	RB
	LOB	XNOT
	C7	XX
ZZ0	Penn	NIL - no equivalent tag
	LOB	ZZ
	C7	ZZ1,ZZ2
ZZ2	Penn	NIL - no equivalent tag
	LOB	ZZ
	C7	ZZ1,ZZ2
PRE,PRE- This is a CUVPlus tag for prefixes	Penn	NIL - no equivalent tag
	LOB	NIL - no equivalent tag
	C7	NIL - no equivalent tag

Appendix 2: ProPOSEL’s Software Tools

A2.1. Introduction

The prosody lexicon comes as a textfile with each of its 104,049 entries presented as a series of pipe-separated fields:

```
carsick|AJ0|0|’kAsIk|OA%|AJ0:-1|2|12|JJ|C|JJ,JJB,JNP|JJ,JK|
'k#-"sIk|'k#:1 "sIk:2|[CVV][CVC]
```

The contents of each field are as follows.

Field	Contents	Example
1	wordform	`carsick`
2	C5 PoS tag	`AJ0`
3	capitalization flag	`0`
4	SAM-PA phonetic transcription	`‘kAsIk`
5	CUV2 tag plus frequency rating	`OA%`
6	C5 PoS tag plus frequency rating	`AJ0:-1`
7	syllable count	`2`
8	lexical stress pattern	`12`
9	Penn Treebank PoS tag(s)	`JJ`
10	CFP tag	`C`
11	LOB PoS tag(s)	`JJ,JJB,JNP`
12	C7 PoS tag(s)	`JJ,JK`
13	DISC syllabified transcription	`'k#-"sIk`
14	syllable-stress mapping	`'k#:1 "sIk:2`
15	CV pattern	`[CVV][CVC]`

Table A.1: Fields in ProPOSEL

This format, plus the contents of field 1 and fields 3 – 7, are derived from two parent files: Roger Mitton’s computer-usable dictionary CUV2 (Mitton, 1992) and Jennifer Pedler’s updated version of same (Pedler and Mitton, 2003). This tutorial mainly uses information from newly created *prosodic-syntactic* fields: field 2 and fields 8 - 15 and refers the reader to a full account of lexicon build in Chapter 5 of this thesis. The code has been updated to NLTK version 0.9.8, and the code here is compatible with the version of NLTK used in the recently published book (Bird *et al.*, 2009).

A2.2. Preparing the prosody lexicon for NLP

We will initially just use a *sample* from the prosody lexicon for purposes of illustration. The lexicon can be read in as a list of string entries via a two step process (**Listing A.1**):

```
# import latest version of NLTK
import nltk, re, pprint

# strings terminate in newline \n
lexicon = open('filepath', 'rU').readlines()

# strip away \n character
lexicon = map(string.strip, lexicon)
```

Listing A.1: Reading in and tokenizing ProPOSEL (Method 1)

Individual fields can then be tokenized with an additional line of code.

```
# tokenize at pipe symbols
lexicon = [line.split('|') for line in lexicon]
>>> lexicon
[.., ['carsick', 'AJ0', '0', "'kAsIk", 'OA%', 'AJ0:-1', '2', '12',
'JJ',  'C',  'JJ,JJB,JNP',  'JJ,JK',  "'k#-,sIk",  "'k#:1  ,sIk:2",
'[CVV][CVC]',],..]
```

Listing A.1: continued.

A2.2.1. Alternative for tokenizing entries and fields in the lexicon

An alternative way of converting the lexicon textfile into a list of lists, where the contents of each entry field appear as separate tokens, is given in **Listing A.2**.

```
# import latest version of NLTK
import nltk, re, pprint
```

```
# instantiate a LineTokenizer()
tokenizer = LineTokenizer()


# read in the lexicon
lexicon = open('filepath', 'rU').read()


# use NLTK’s LineTokenizer Class to convert string entries to
tokenized lines
lexicon = [line.split('|') for line in
list(tokenizer.tokenize(lexicon))]
```

Listing A.2: Reading in and tokenizing ProPOSEL (Method 2)

Either way, the result is the same. If we now use the Python built-in `enumerate()` function on our *sample* lexicon, we can view the output - as in `lexicon[20]` (**Listing A.3**).

```
>>> for index, value in enumerate(lexicon):
    print index, value
20 ['carrying', 'VVG', '0', "'k&rIIN", 'Jb%', 'VVG:58', '3', '100',
'VBG', 'C', 'VBG', 'VVG,VVGK', "'k{-rI-IN", "'k{:1 rI:0 IN:0",
'[CV][CV][VC]']
```

Listing A.3: Inspecting tokenized entry fields

A.2.2 Selecting specific fields in the lexicon

Depending on our purpose, we may only need certain information in the lexicon. The first line of code in the next listing creates a new lexicon called `syllables` where entries consist of: wordform, PoS tag, syllable count and syllable-stress mapping. We can then view this cut down version of the lexicon.

```
syllables = [[line[0], line[1], line[6], line[13]] for line in
lexicon]
>>> for line in syllables:
      print ' '.join(line[:])
# best printed to file with whole lexicon!
carrier-pigeons NN2 5 'k{:1 r7:0 ,pI:2 _Inz:0
carriers NN2 3 'k{:1 r7z:0
carries VVZ 2 'k{:1 rIz:0
```

Listing A.4: Selecting specific fields in the lexicon

Output from **Listing A.4** reveals some anomalies in the lexicon. The resulting entries for *carrier-pigeons* and *carriers* display information on syllable count at variance with the syllable-stress mapping. Such anomalies are deliberate, in a sense. The prosody lexicon was created from various sources and one of the insights gained through this process is that in English, syllabification - like other aspects of prosody - is often a matter of choice and natural discrepancies arise between one native speaker and another (*cf*. Chapter 6.4.3 and 6.4.4). As a general rule, the syllable count in field 7 of the prosody lexicon constitutes a subjective choice or judgement by a native speaker while the pronunciation forms in fields 13 and 14, plus the syllabified CV patterns in field 15 are more *canonical*. It is up to the user to decide how to negotiate such variance.

A.2.3 Using Python's set() method to capture attribute-value mappings

The prosody lexicon was originally intended as a prosodic annotation tool for machine learning, to be used in conjunction with tagged speech corpora and this is discussed in Section A.5 below. Here we may simply note that using Python's `set()` method on a single field in the lexicon retrieves the set of all possible values for that field. Thus, if *lexical stress pattern* is interpreted as a potential classificatory feature for a given machine learning task, we can use the following line of code on field 8 of our lexicon to obtain all permutations for this feature. This code uses the field selector in **Listing A.4** in a Python list comprehension as single argument to the `set()` method.

```
lexStressValues = list(set([(line[7]) for line in lexicon]))

>>> lexStressValues
```

```
['201', '10', '12', '1020', '120', '1 0', '1', '2001', '100'] #
sample set only
```

Listing A.5: Inspecting attribute-value mappings within lexicon fields

Listing A.5 again shows an anomaly - this time resulting from choices made during lexicon build - in that this set (from the sample lexicon – the full set is much larger) contains two identical lexical stress patterns: `{'10'}` and `{'1 0'}` where the latter includes a whitespace character. Instances of this kind originally arose from a decision to preserve variance in lexical stress patterns derived from two different sources (Chapter 6.4.3). Again, it is left to the user to determine how to accommodate such distinctions.

A.3 Mapping variant syntactic information in the prosody lexicon

The prosody lexicon incorporates alternative PoS tagging schemes: C5, Penn Treebank, LOB and C7 (*cf.* Appendix 1 for details of the mapping between schemes in ProPOSEL); and it is possible to map between them via a one-step process similar to that used in **Listing A.5**. We may, for example, wish to map C5 to Penn and print this out in tabulated format. **Listing A.6** uses a declarative style - again via list comprehension as argument to Python's `set()` method - to obtain the mapping and then wraps up the formatting in a function.

```
mapTags = list(set([(line[1], line[8]) for line in lexicon]))
>>> def getMapping(mapTags):
     print '%s %20s\n' % ('C5 ', 'Penn Treebank') # creates header
row
     for line in mapTags:
        print '%s %20s' % (line[0],line[1])    # cf. NLTK Book 6.3.2









>>> getMapping(mapTags)
C5          Penn Treebank
```

```
NN2                 NNS          #   Note:   this   mapping   is
                                 incomplete
VVI                  VB
                                 # because only a sample from the
AJ0                  JJ
                                 # lexicon is used for this demo
VVB              VB,VBP

VVG                 VBG

VVZ                 VBZ

NP0            NNP,NNPS

NN1                  NN
```

Listing A.6: Obtaining simplified mapping of variant PoS schemes in ProPOSEL

The user may note from this small sample in **Listing A.6** that there are instances where the source tag does not map neatly onto one target tag and this problem is compounded when the tagset(s) involved are less sparse - LOB, for example, or C7. There is also the question of linguistic interpretation. Ambivalence surrounding infinitive and base forms of lexical verbs is evident here: C5 distinguishes between the two - `<VVI>` versus `<VVB>` - whereas the Penn Treebank has one tag for 'base form' `<VB>` and then another tag `<VBP>` for non-3sg (not 3rd. person singular) present tense.

A.3.1 Dealing with enclitics, Saxon genitives and one-to-many mappings

In this section we will use information from the entire prosody lexicon for demonstration. The following code - familiar from Section A.2 - maps the set of all C5 PoS tags in the lexicon to their equivalent symbolic values in LOB.

```
import nltk, re, pprint, copy, itertools, string

lexicon = open('filepath', 'rU').read()

# the complete prosody lexicon

lexicon = lexicon.splitlines()

lexicon = [line.split('|') for line in lexicon]

mapTags = list(set([(line[1], line[10]) for line in lexicon]))

>>> len(mapTags)

96
```

Listing A.7: The set of C5 to LOB mappings in the lexicon

The C5 tagset contains 62 part-of-speech tags, including 4 tags for punctuation. The set of 96 C5 tags in the lexicon, evident from **Listing A.7**, includes enclitics and possessive forms like ***"I'll"*** `<PNP+VMO>` and ***"Lloyd's"*** `<NPO+POS>`. The variable `mapTags` also reveals that around 41% (39 out of 96) of these C5 to LOB mappings are one-to-many. **Table A.2** below provides the following example strings from mapTags where the problem emerges of separately tokenizing the variant LOB tags, plus their possessive or enclitic attachments, to match the corresponding C5 token.

Example strings from mapTags with C5 token given first	**Possessives and enclitics need attaching to all PoS variants in LOB**
'NP0**+POS**', 'NP,NPL,NPLS,NPS,NPT,NPTS**+$**'	Symbolic mapping of **Saxon genitive** in C5 and LOB
'DTQ**+VM0**', 'WDT,WQL,WP,WP$**+MD**'	One-to-many mapping for **wh-determiner** with encliticised **modal**
'PNP**+VBB**', 'PP3AS,PP3O,PP3OS,PP$$,PP1A,PP1AS, PP1O,PP1OS,PP2,PP3,PP3A**+BEM,BER**'	One-to-many mapping for both **personal pronoun** and encliticised base form of **BE**

Table A.2: Problems with genitives, enclitics and one-to-many mappings

A.3.1.1 What is our target format?

We have so far been working with the C5 and LOB tagsets but code listings in this section can be adapted for other tagging schemes in the lexicon. Our demonstration target here is to map *combination* C5 tokens to a series of equivalent *combination* tokens in LOB as follows.

C5 combo	**Possible corresponding LOB combos**
'NP0**+POS**',	'NP**$**', 'NPL**$**', 'NPLS**$**', 'NPS**$**', 'NPT**$**', 'NPTS**$**'
'DTQ**+VM0**',	'WDT**+MD**', ‘WQL**+MD**', 'WP**+MD**', 'WP$**+MD**'
'PNP**+VBB**',	'PP3AS**+BEM**', 'PP3O**+BEM**', 'PP3OS**+BEM**', 'PP$$**+BEM**', 'PP1A**+BEM**', 'PP1AS**+BEM**', 'PP1O**+BEM**', 'PP1OS**+BEM**', 'PP2**+BEM**', 'PP3**+BEM**', 'PP3A**+BEM**', 'PP3AS**+BER**', 'PP3O**+BER**', 'PP3OS**+BER**', 'PP$$**+BER**', 'PP1A**+BER**', 'PP1AS**+BER**', 'PP1O**+BER**', 'PP1OS**+BER**', 'PP2**+BER**', 'PP3**+BER**', 'PP3A**+BER**',

Table A.3: Specifiying target format for C5 to LOB mapping

A.3.2.2 Subsuming tag attachments into variant LOB tag tokens

The first step is to separate the *attachment* from the rest of the LOB string (**Listing A.8**).

```
mapTags = list(set([(line[1], line[10]) for line in lexicon]))


tagSplit = [line[1].split('+') for line in mapTags] # line[1] holds
```

```
the LOB tags


>>> tagSplit
[..['NP,NPL,NPLS,NPS,NPT,NPTS', '$'],..] # example output
```

Listing A.8: Re-formatting LOB tokens: initial step

The variable `tagSplit` can then be used and transformed within interdependent reformatting functions (**Listing A.9**) which need to handle enclitics separately from Saxon genitives. This staged reformatting is explained in Section A.3.2.3 but readers should also look carefully at comment lines in **Listing A.9** and also try things out for themselves.

A.3.2.3. High-level description

(1) Instantiate an empty list to store transformations created during reformatting.

(2) Incorporate a plus sign in the tokenized LOB string in `tagSplit` as prefix to each individual PoS tag representing an enclitic.

(3) Apply the transformations in step (2). Loop over the latest version of `tagSplit` and push each item onto the stack in our empty list while simultaneously ensuring that the `<$>` tag denoting Saxon genitive is incorporated within LOB tokens where necessary.

(4) Loop over the result of step (3) and tokenize each individual PoS in each string of LOB tags.

(5) Loop over the result of step (4). Attach enclitics to each LOB token and pop any unwanted material.

(6) Apply steps (1) to (5) by calling the reformatting function in step (5).

(7) Unpack unwanted structure so that the final sequence is a list of lists where each index comprises C5 token and equivalent LOB tokens.

A.3.2.4 Reformatting functions

```
import nltk, re, pprint, copy, itertools, string
from nltk.tokenize import *
tokenizer = WhitespaceTokenizer() # tokenization via whitespace as
separator
```

```
lexicon = open('filepath', 'rU').read()
lexicon = lexicon.splitlines()
lexicon = [line.split('|') for line in lexicon]

mapTags = list(set([(line[1], line[10]) for line in lexicon]))
tagSplit = [line[1].split('+') for line in mapTags]

empty = [] # somewhere to store list transformations created during
reformatting


def format1(tagSplit):
    """
    Formatting function dealing only with single instance
enclitics.
    Restores '+' as prefix as in: [..['WDT,WQL,WP,WP$', '+MD'],..]
    """
    for line in tagSplit: # next condition excludes genitives
        if len(line) > 1 and len(line[1]) > 1:
            line[1] = re.sub(line[1], '+' + line[1], line[1])


def format2(tagSplit): # argument is transformed tagSplit from
previous function
    """
    Formatting function dealing with multiple instance enclitics.
    Restores '+' as prefix as in: [..['...PP3A', '+BEM,+BER'],..]
    Takes as argument list transformation created from previous
function.
    """

    for line in tagSplit:
        if len(line) > 1:
            line[1] = re.sub(',', ',+', line[1])# cases like ‘+BEM,BER’


def getEnclitics(tagSplit, empty): # this reformats Saxon genitives
    """
    Formatting function which takes as argument list transformation
created by calling previous two functions.  It loops over this
latest version of tagSplit and appends each item to container list
(empty) while simultaneously ensuring that the <$> tag denoting
Saxon genitive is incorporated within LOB tokens where necessary.
    """

    format1(tagSplit) # applies previous function
    format2(tagSplit) # applies previous function
    for line in tagSplit: # transformed tagSplit via format1 and format2
        if len(line) == 1: # all lines without enclitics or Saxon genitives
            empty.append(line)
        elif len(line) > 1: # only do this for enclitics or genitives
            if line[1] == '$': # treat genitives separately from enclitics
                line[0] = line[0] + ',' # adds a trailing comma to enable…
                line = re.sub(r',', '$,', line[0])
# Previous line adds a trailing comma to enable last item to pick up '$'
                line = line[:-1] # chops off trailing '$,'
                empty.append([line]) # preserves structure of nested list
            else:
                empty.append(line) # only applies to enclitics
```

```
def format3(empty): # tokenizes each individual PoS tag
    """
    Formatting function which tokenizes all LOB tags via whitespace
separator. Takes as argument list transformation built via previous
function. Enclitics still consist of two separate tokens after
calling this function as in:
[..[['WP', 'WPA', 'WPO', 'WP$', 'WP$R', 'WPOR', 'WPR'],
['+MD']],..]
    """
    getEnclitics(tagSplit, empty) # applies previous function & builds list
    for index in empty: # loop tokenizing is via whitespace replacing commas
        if len(index) == 1:
            index[0] = list(tokenizer.tokenize(re.sub(',', ' ', index[0])))
        elif len(index) > 1:
            index[0] = list(tokenizer.tokenize(re.sub(',', ' ', index[0])))
            index[1] = list(tokenizer.tokenize(re.sub(',', ' ', index[1])))

def format4(empty): # attaches enclitics
    """
    Formatting function which merges PoS tags in enclitics.  This
function call applies all the transformations on tagSplit described
in the previous functions.  Resulting structure still needs to be
unpacked.
    """
    format3(empty)  # applies previous function
    for line in empty: # loops over transformed version of empty
        if len(line) > 1:
            if len(line[0]) == 1 and len(line[1]) == 1: # e.g.
[['MD'], ['+XNOT']]
                for x, y in itertools.izip(line[0], line[1]):
# Previous line combines tags
                    line.append([x + y]) # appends combined tags
                    del line[0:2] # removes separate tags
            elif len(line[0]) > 1 and len(line[1]) >= 1:
# Previous line is for multiple variants
                for x in line[0]: # loops over PoS variants
                    for y in line[1]: # loops over enclitics
                        line.append([x + y])
# Outputs from previous line e.g. ['WPO+MD'],['WP$+MD']]
                del line[0:2]#removes separate tags on combining enclitics

# All the above functions can then be called via a single line of
code:

format4(empty) # applies all the transformations

# We now have LOB combination tokens (cf. Table A1).  What remains
to be done is unpack this deeply nested structure, for example:

[..[['PP3AS+BEM'], ['PP3AS+BER'], ['PP3O+BEM'], ['PP3O+BER'],
['PP3OS+BEM'], ['PP3OS+BER'], ['PP$$+BEM'], ['PP$$+BER'],
['PP1A+BEM'], ['PP1A+BER'], ['PP1AS+BEM'], ['PP1AS+BER'],
['PP1O+BEM'], ['PP1O+BER'], ['PP1OS+BEM'], ['PP1OS+BER'],
['PP2+BEM'], ['PP2+BER'], ['PP3+BEM'], ['PP3+BER'], ['PP3A+BEM'],
['PP3A+BER']],..]
```

Listing A.9: Re-formatting functions

A.3.3 Unpacking unwanted structure and printing out a mapping

Enclitics in the resulting list sequence object `empty` are still too deeply nested and we need to unpack them. To do this in one step, we need to treat indices containing LOB combination tags differently than others. **Listing A.10** offers one solution and also provides a neatly aligned printout of the C5 to LOB mapping obtained from the prosody lexicon as an illustration.

```
empty2 = [] # instantiates a new container

for line in empty:
    if len(line) == 1: # for indices which don’t contain enclitics
        empty2.append(line[0])
    elif len(line) > 1: # in the case of enclitics
        empty2.append([' '.join(index) for index in line])

>>> empty2

# Values in this new container are identical in structure to
ProPOSEL’s keys:

('VBB+XX0', ['BEM+XNOT', 'BER+XNOT'])
('NP0', ['NP', 'NPL', 'NPLS', 'NPS', 'NPT', 'NPTS'])
('VBI', ['BE'])
# Obtaining a mapping:

tagsC5 = [index[0] for index in mapTags]
# set of all C5 tags in prosody lexicon

tagsC5LOB = zip(tagsC5, empty2) # obtains mapping C5 > LOB

# Obtaining an example printout:

print 'C5       :  LOB\n' # header row

for index in tagsC5LOB:
    if len(index[0]) > 3: # if index[0] looks like: 'VBB+XX0'
        print index[0], ' : ', ' '.join(index[1])
    elif len(index[0]) <= 3: # if index[0] isn’t an enclitic
        print index[0], '     : ', ' '.join(index[1])
# leave some extra space

# Aligned printout looks like:

C5       :  LOB

VBB+XX0  :  BEM+XNOT BER+XNOT
NP0      :  NP NPL NPLS NPS NPT NPTS
VBI      :  BE
```

Listing A.10: Unpacking unwanted structure

A.4. Using the lexicon as a prosodic annotation tool

We now have an object - `tagsC5LOB` - which contains one-to-many mappings of C5 tokens to an array of equivalent LOB tokens, including *combination* tokens for enclitics. Code listings in sections 4 and 5 of this tutorial utilise this object, rather than the raw information in the lexicon textfile (e.g. field 11 for LOB), to match incoming corpus text in the form of (token, tag) tuples to entries in the lexicon and automatically annotate that text with additional prosodic information. **Listing A.11** uses a short sentence from Section C of the LOB-tagged Spoken English Corpus and enriches the existing annotation with prosodic information on: syllable count (lexicon field 7); lexical stress pattern (field 8); CFP status (field 10); and the distribution of stressed and unstressed syllables represented in DISC format (field 14). Readers should note that this tutorial does not deal with punctuation and such tokens will not accumulate additional information.

```
text  = ['both', 'ABX'), ('propositions', 'NNS'), ('are', 'BER'),
('false', 'JJ')]
text2 = [list(line) for line in text] # lists are mutable

for line in text2:
    for index in tagsC5LOB: # the mapping object created in Listing A10 above
        for i in range(len(index[1])):
            if line[1] == index[1][i]: # if LOB tags match
                line.append(index[0])  # add the C5 tag

>>> text2 # tokens now displaying LOB and C5 tags
[['both', 'ABX', 'DT0'], ['propositions', 'NNS', 'NN2'], ['are',
'BER', 'VBB'], ['false', 'JJ', 'AJ0']]

# Create a new version of the lexicon with desired fields only
lexicon2 = [(line[0], line[1], line[6], line[7], line[9], line[13])
for line in lexicon]

"""
The following loop only deals with non-ambiguous matches, searching
for a match between wordform and C5 tag in the two objects: text2
and tagsC5LOB.  Ambiguous matches will be further discussed – see
4.2 below.
"""
for line in text2:
    for entry in lexicon2:
        if len(line) == 3: # non-ambiguous matches only
            if line[0] == entry[0] and line[2] == entry[1]:
                line.append(entry[2])
                line.append(entry[3])
                line.append(entry[4])
                line.append(entry[5])

>>> for line in corpusText2: # result for illustrative fragment
      print line

['both', 'ABX', 'DT0', '1', '1', 'F', "'b5T:1"]
['propositions', 'NNS', 'NN2', '4', '2010', 'C', '"prQ:2 p@:0 'zI:1
SHz:0']
['are', 'BER', 'VBB', '1', '1', 'C', "'#R:1"]
['false', 'JJ', 'AJ0', '1', '1', 'C', "'f$ls:1"]
```

Listing A.11: Annotating corpus sample via ProPOSEL

A.4.1 Discussion of ambiguous cases

So far, we have encountered one-to-many mappings where one C5 tag equates to two or more LOB variants; **Listing A.11** can deal with this eventuality. However, a *backfiring* problem emerges for those few instances where the one-to-many mapping works the other way round. Prepositions, for example, have one tag in LOB but one of two tags in C5, which isolates `('of', 'PRF')` from all other prepositions: `<'PRP'>`. Using a longer extract (85 tokens in all) from the same source, which we will call `text3`, we can check outputs from **Listing A.11** to see how many of them have successfully accumulated extra prosodic features. How many have been missed and why have they been missed? To answer this, we need to inspect the transformed object `text3_transformed`.

```
text3_transformed = [list(line) for line in text3] # lists are
mutable

for line in text3_transformed:
    for index in tagsC5LOB: # mapping object created in Listing A10
        for i in range(len(index[1])):
            if line[1] == index[1][i]: # if LOB tags match
                line.append(index[0])  # add the C5 tag

>>> len(text3_transformed)
85

# Sample outputs show why some indices have not been tagged - see
comments

['I', 'PPSS'] # untagged because word is in upper case
['want', 'VB', 'VVI', 'VVB'] # one-to-many mapping LOB < C5
['to', 'TO', 'TO0', '1', '1', 'F', "'tu:1"]
['enlarge', 'VB', 'VVI', 'VVB']
['on', 'IN', 'PRF', 'PRP']
['the', 'ATI', 'AT0', '1', '1', 'F', "'Di:1"]
['contrast', 'NN', 'NN1', '2', '10', 'C', "'kQn:1 tr#st:0"]
['between', 'IN', 'PRF', 'PRP']
['DIYE', 'NP', 'NP0'] # not in ProPOSEL lexicon
['and', 'CC', 'CJC', '1', '1', 'F', "'{nd:1"]
['economic', 'JJ', 'AJ0', '4', '2010', 'C', '"i:2 k@:0 'nQ:1
mIk:0']
['orthodoxy', 'NN', 'NN1', '4', '1000', 'C', "'$:1 T@:0 dQk:0
sI:0"]

tagged = [index for index in text3_transformed if len(index) == 7]
# total tagged
untagged = [index for index in text3_transformed if len(index) <=
4] # total missed
variants = [index for index in text3_transformed if len(index) ==
4] # LOB < C5
misc = [index for index in text3_transformed if len(index) <= 3]
# miscellaneous
>>> len(tagged)
42
>>> len(untagged)
43
>>> len(variants)
23
>>> len(misc)
20
```

Listing A.12: Inspecting annotation outputs

We have successfully annotated about 50% of our extract. One-to-many mappings in the direction LOB < C5 account for about 50% of untagged data; we will fix this problem with a *patch* in Section 6.5, where we also implement the prosody lexicon as a Python dictionary. There are the remaining untagged indices in `misc` **(Listing A.12)** that cannot be accounted for in this way. The interested reader will find that *misc* contains capitalised items and proper nouns `['MacQuedy', 'NP', 'NP0']`, abbreviations `['DIYE', 'NP', 'NP0']`, compounds `['do-it-yourself', 'JJB', 'AJ0']` and adverbials `['first', 'RB', 'AV0']`. In the case of adverbials, it may well be that they are tagged differently in the corpus than in the original lexicon entry: CUVPlus classes *'first'* as an ordinal `<ORD>` for example.

A.5. Implementing the prosody lexicon as a Python dictionary

The Python programming language has a dictionary mapping object with entries in the form of (key, value) pairs. Each key must be unique and immutable (e.g. a *string* or *tuple*), while the values can be any type (e.g. a *list*). This syntax can be exploited when transforming the prosody lexicon into a Python dictionary. Tuples can be used to create compound lookup keys comprising wordform and PoS tag which in turn are associated with multiple values in the form of a list of tokens from selected fields for any given entry. Thus, using a sample of 9 entries to represent our lexicon and version 0.9.8 of NLTK, we can transform it into a Python dictionary or associative array.

```
import nltk, re, pprint
from nltk.tokenize import *
tokenizer = LineTokenizer()
lexicon = """
cascade|NN1|0|k&'skeId|I2%,K6%|NN1:2|2|01|NN|C|NN1,NNT1,NNU1,ND1|NR
,NN,NNP
cascade|VVB|0|k&'skeId|I2%,K6%|VVB:0|2|01|VB,VBP|C|VV0|VB
cascade|VVI|0|k&'skeId|I2%,K6%|VVI:0|2|01|VB|C|VVI|VB
cascaded|VVD|0|k&'skeIdId|Ic%,Id%|VVD:1|3|010|VBD|C|VVD|VBD
cascaded|VVN|0|k&'skeIdId|Ic%,Id%|VVN:0|3|010|VBN|C|VVN,VVNK|VBN
cascades|NN2|0|k&'skeIdz|Ia%,Kj%|NN2:1|2|01|NNS|C|NN2,NNJ2,NNT2,NNU
2,NNO2|NNS,NRS,NNPS,NNUS
cascades|VVZ|0|k&'skeIdz|Ia%,Kj%|VVZ:-1|2|01|VBZ|C|VVZ|VBZ
cascading|VVG|0|k&'skeIdIN|Ib%|VVG:1|3|010|VBG|C|VVG,VVGK|VBG
cascading|AJ0|0|k&'skeIdIN|Ib%|AJ0:0|3|010|JJ|C|JJ,JK|JJ,JJB,JNP
"""

lexicon = [line.split('|') for line in list(tokenizer.tokenize(lexicon))]
lexKeys = [(index[0], index[1]) for index in lexicon]
```

```
# dictionary lookup keys

lexValues = [[index[6], index[7], index[9]] for index in lexicon]
# values

buildDict = dict(zip(lexKeys, lexValues))



>>> buildDict

{('cascades', 'NN2'): ['2', '01', 'C'], ('cascaded', 'VVN'): ['3',
'010',  'C'],  ('cascade',  'VVB'):  ['2',  '01',  'C'],  ('cascade',
'NN1'): ['2', '01', 'C'], ('cascading', 'VVG'): ['3', '010', 'C'],
('cascaded', 'VVD'): ['3', '010', 'C'], ('cascade', 'VVI'): ['2',
'01', 'C'], ('cascades', 'VVZ'): ['2', '01', 'C'], ('cascading',
'AJ0'): ['3', '010', 'C']}
```

Listing A.13: Transforming the prosody lexicon into a Python dictionary

This returns an as yet unsorted dictionary. To reorder items and inspect this series of linguistic observations on wordform and part-of-speech mapped to syllable count, lexical stress pattern and content/function word status, we can use the following code.

```
>>> jumble = buildDict.keys()
>>> def sortIt(jumble):
     jumble.sort()
     for k in jumble:
                          print ' '.join(k), ' '.join(buildDict[k])

>>> sortIt(jumble)
cascade NN1 2 01 C
cascade VVB 2 01 C
cascade VVI 2 01 C
cascaded VVD 3 010 C
cascaded VVN 3 010 C
cascades NN2 2 01 C
cascades VVZ 2 01 C
cascading AJ0 3 010 C

cascading VVG 3 010 C
```

Listing A.14: Sorting a Python dictionary

A.5.1 Intersection between the transformed lexicon and corpus text

The compound keys (wordform, C5 PoS tag) in our transformed prosody lexicon facilitate linkage with speech corpora, especially if the corpus is tagged with C5 like the BNC. Incoming corpus text - also in the form of (token, tag) tuples - can be matched against dictionary keys; and thus intersection enables text to accumulate additional prosodic annotations which constitute potential features for machine learning tasks. CFP status, for example - field 10 in the lexicon - has proved a very effective attribute for automatic phrase break prediction (Liberman and Church, 1992; Busser *et al.*, 2001).

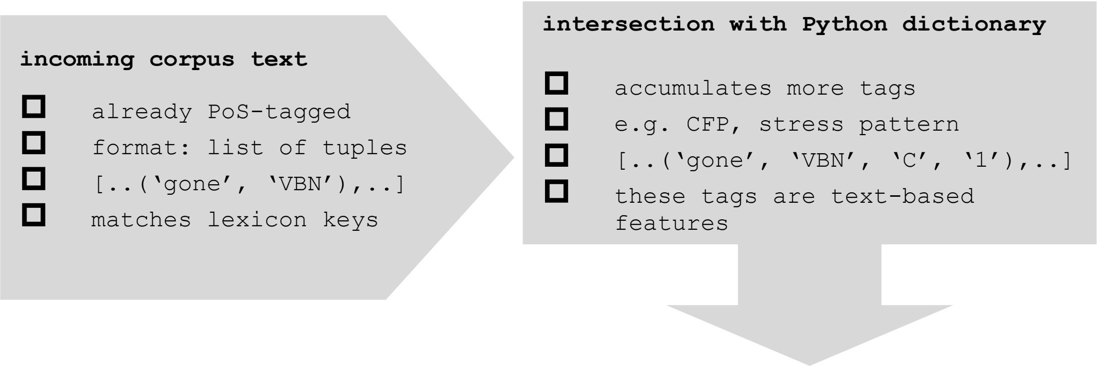


Figure A.1: Input text intersects with Python dictionary keys and acquires additional tags from the prosody lexicon

If the incoming corpus text is tagged with a different scheme, we are still able to use the transformed lexicon as a text annotation tool; as in Section 6.4, the object `tagsC5LOB` (containing one-to-many mappings of C5 tokens to an array of equivalent LOB tokens in the desired format) will be used to promote dictionary lookup.

A.5.2 Patching the backfiring problem

The object `variants` in code **Listing A.12** reveals a problem with one-to-many mappings in the unexpected direction C5 < LOB. It contains items such as prepositions, subordinating conjunctions, infinitives/base forms of verbs (*cf.* discussion in section A.3), present and past participles, and WH-pronouns; a brief explanation of why these instances occur is included in the comments below.

```
>>> for line in variants:
      print line            # example outputs contain items like
['of', 'IN', 'PRF', 'PRP']
# C5 has 2 tags for prepositions, one being unique to 'of'

['that', 'CS', 'CJT', 'CJS']
# C5 has 2 tags for subordinating conjunctions, one being unique to'that'

['pay', 'VB', 'VVI', 'VVB']
# C5 distinguishes between the infinitive and base form of the verb

['used', 'VBN', 'VVN', 'VDN'] # C5 has a separate tag for 'done'

['unreflecting', 'VBG', 'VVG', 'VDG'] # C5 has separate tag for 'doing'

['which', 'WP', 'PNQ', 'DTQ']
# patch will only use <DTQ> since this is how 'which' is tagged in CUVPlus
```

Listing A.15: Inspecting one-to-many annotation outputs

Fortunately, there is only a small number of one-to-many mappings from C5 < LOB and therefore we can solve this problem, by and large, with the following patch - though readers should note it may not be exhaustive, simply because it has not yet been tested on a sufficient amount of corpus text. Readers should also note that we have gone back to the object `text3_transformed` *(cf.* ***Listing A.12****) before* intersection with the lexicon.

```
text3_transformed = [list(line) for line in text3] # lists are
mutable

for line in text3_transformed:
    for index in tagsC5LOB: # mapping object created in Listing A10
        for i in range(len(index[1])):
            if line[1] == index[1][i]: # if LOB tags match
                line.append(index[0])  # add the C5 tag

# HERE IS THE PATCH - WITH SOME EXPLANATIONS GIVEN IN COMMENTS:

for index in text2:
    if len(index) == 4: # if there are 2 equivalent C5 tags
        if index[0] == 'of':
            index.remove('PRP')
        elif index[0] != 'of' and index[1] == 'IN':
            index.remove('PRF')
        elif index[0] == 'that':
            index.remove('CJS')
        elif index[0] != 'that' and index[1] == 'CS':
            index.remove('CJT')
        elif index[0] == 'done':
            index.remove('VBN')
        elif index[1] == 'VBN' and index[0] != 'done':
            index.remove('VDN')
        elif index[0] == 'doing':
            index.remove('VBG')
        elif index[1] == 'VBG' and index[0] != 'doing':
            index.remove('VDG')
        elif index[0] == 'which':
            index.remove('PNQ') # retain tag in original documentation
        elif index[1] == 'BE':
            index.remove('VBB')  # appears less frequently in the
lexicon
        elif index[1] == 'HV':
            index.remove('VHI')  # appears less frequently in the
lexicon
        elif index[1] == 'DO':
            index.remove('VDI')  # appears less frequently in the
lexicon
        elif index[1] == 'VB' and index[0] not in ['be', 'do', 'have']:
            index.remove('VVI')  # appears less frequently in the
lexicon

>>> variants # if we inspect the list of one-to-many LOB > C5 mappings…

[] # …we find that it’s empty
```

Listing A.16: Patch for resolving one-to-many mappings

A.5.3. High level description of dictionary lookup

(1) Use the first two fields of the prosody lexicon in a tuple to create the immutable dictionary keys.

(2) Select corresponding values from the remaining fields in the lexicon.

(3) Build a Python dictionary from these compound keys and value arrays.

(4) Ensure all indices in the object `text3_transformed` are of equal length.

(5) Isolate the (word, C5 tag) tokens in `text3_transformed` ready for intersection with dictionary keys.

(6) Loop through the two iterables - i.e. the dictionary keys and the (word, C5 tag) tokens in `text3_transformed` - in parallel, using Python's itertools() module. If there is a match, then append the value array associated with that key to the index in `text3_transformed`.

(7) Print the result to file in a format of your choice.

A.5.4. Code listing for dictionary lookup

```
lexKeys = [(index[0], index[1]) for index in lexicon] # Step (1)

lexValues = [[index[6], index[7], index[9], index[13]] for index in lexicon] # Step
(2)

buildDict = dict(zip(lexKeys, lexValues)) # Step (3)

for index in text3_transformed:
     if len(index) == 2:
         index.append('None') # Step (4)

match = [(index[0], index[2]) for index in text3_transformed] # Step (5)

for x, y in itertools.izip(match, text3_transformed): # Step (6)
    if x in buildDict.keys(): # if tuple matches dictionary keys
        y.append(buildDict[x]) # append value array to index in corpusText2

    else:
        y.append('No_match')

# EXAMPLE RESULT FROM STEP 7, WITH FORMATTING APPLIED:

wordform:             individual
PoS tag:              JJ
syllable count:       5
stress pattern:       2010
CFP tag:              C
stress distribution: ,In:2 dI:0 'vI:1 _9l:0
```

```
wordform:            willingness
PoS tag:             NN
syllable count:      3
stress pattern:      100
CFP tag:             C
stress distribution: 'wI:1 lIN:0 nIs:0

wordform:            to
PoS tag:             TO
syllable count:      1
stress pattern:      1
CFP tag:             F
stress distribution: 'tu:1

wordform:            pay
PoS tag:             VB
syllable count:      1
stress pattern:      1
CFP tag:             C
stress distribution: 'p1:1

wordform:            should
PoS tag:             MD
syllable count:      1
stress pattern:      1
CFP tag:             F
stress distribution: 'SUd:1

wordform:            be
PoS tag:             BE
syllable count:      1
stress pattern:      1
CFP tag:             C
stress distribution: 'bi:1

wordform:            the
PoS tag:             ATI
syllable count:      1
stress pattern:      1
CFP tag:             F
stress distribution: 'Di:1

wordform:            main
PoS tag:             JJB
syllable count:      1
stress pattern:      1
CFP tag:             C
stress distribution: 'm1n:1

wordform:            test
PoS tag:             NN
syllable count:      1
stress pattern:      1
CFP tag:             C
stress distribution: 'tEst:1
```

Listing 6.17: Dictionary lookup and final annotation outputs

A.6. Concluding comments

This stand-alone software tutorial as a guide to using ProPOSEL has been written in the style of the NLTK online book, with step-by-step, fully commented code, and is another aspect of language resource creation as output from this thesis. Much of the code, particularly from Sections A.3 to A.5, has also been instrumental in preparing and annotating datasets used in succeeding chapters, including ProPOSEC (§ 8.10).

Appendix 3: ADTree alternating decision tree classifier models

ADTree Run 5: 31 features, including punctuation uses fine-grained syntactic information defined in this thesis (§9.4)

```
=== Classifier model (full training set) ===

Alternating decision tree:

: -0.663
|  (1) punct = nonterminal: -0.354
|  |  (3) postpos1 = noun: -1.161
|  |  (3) postpos1 != noun: 0.175
|  |  |  (5) postpos1 = preposition: 0.358
|  |  |  (5) postpos1 != preposition: -0.108
|  |  |  (10) postpos1 = conjunctionTHAT: 0.91
|  |  |  (10) postpos1 != conjunctionTHAT: -0.038
|  (1) punct != nonterminal: 3.69
|  (2) pos = noun: 0.684
|  (2) pos != noun: -0.426
|  |  (4) beat = yes: 0.355
|  |  (4) beat = no: -0.446
|  |  |  (9) pos = adverb: 0.828
|  |  |  (9) pos != adverb: -0.064
|  |  (6) pos = preposition: -0.812
|  |  (6) pos != preposition: 0.068
|  (7) pos = pronounObject: 1.325
|  (7) pos != pronounObject: -0.023
|  |  (8) postpos1 = conjunction: 0.598
|  |  (8) postpos1 != conjunction: -0.033
Legend: -ve = nonbreak, +ve = break
Tree size (total number of nodes): 31
Leaves (number of predictor nodes): 21

Time taken to build model: 0.37 seconds

=== Stratified cross-validation ===
```

ADTree Run 13a: 26 features and no punctuation uses fine-grained syntactic information defined in this thesis (§9.4)

```
=== Classifier model (full training set) ===

Alternating decision tree:

: -0.663
|  (1) pos = noun: 0.677
|  |  (7) postpos1 = prepositionOF: -0.757
|  |  (7) postpos1 != prepositionOF: 0.069
|  (1) pos != noun: -0.437
|  |  (3) jassem = ana: -0.531
|  |  |  (5) pos = adverb: 1.118
|  |  |  (5) pos != adverb: -0.131
|  |  |  |  (6) pos = pronounObject: 1.776
|  |  |  |  (6) pos != pronounObject: -0.139
|  |  |  |  (9) pos = adjectiveArticle: -0.795
|  |  |  |  (9) pos != adjectiveArticle: 0.161
|  |  (3) jassem != ana: 0.342
|  |  (8) pos = preposition: -0.973
|  |  (8) pos != preposition: 0.045
|  (2) postpos1 = noun: -0.86
|  (2) postpos1 != noun: 0.17
|  |  (4) postpos1 = conjunction: 0.755
|  |  (4) postpos1 != conjunction: -0.056
|  |  (10) postpos1 = pronoun: 0.504
|  |  (10) postpos1 != pronoun: -0.046
Legend: -ve = nonbreak, +ve = break
Tree size (total number of nodes): 31
Leaves (number of predictor nodes): 21

Time taken to build model: 0.28 seconds

=== Stratified cross-validation ===
```