Named Entity Recognition from Speech and Its Use in the Generation of Enhanced Speech Recognition Output

Ji-Hwan Kim

Darwin College, University of Cambridge

and

Cambridge University Engineering Department

August 2001

Thesis submitted to the University of Cambridge in partial fulfilment of the requirement for the degree of Doctor of Philosophy

Abstract

The work in this thesis concerns Named Entity (NE) recognition from speech and its use in the generation of enhanced speech recognition output with automatic punctuation and automatic capitalisation. A method for the automatic generation of rules is proposed for NE recognition. Punctuation marks are generated using context and prosody information. Capitalisation is produced based on the results of NE recognition and punctuation generation.

Previous work regarding the NE task is mainly categorised by hand crafted rule-based systems and stochastic systems. By contrast, in this thesis, an automatic rule generating method, which uses the Brill rule inference approach, is proposed. The performance of the rule-based NE recogniser is compared with that of the BBN's commercial implementation called IdentiFinder. When only the sequences of words are available, both systems show almost equal performance as is also the case with additional information such as punctuation, capitalisation and name lists. In cases where input texts are corrupted by speech recognition errors, the performances of both systems are degraded by almost the same level. Although the rule-based approach is different from the widely used stochastic method, these results show that automatic rule inference is a viable alternative to the stochastic approach to NE recognition, while retaining the advantages of a rule-based approach.

A punctuation generation system which incorporates prosodic information along with acoustic and language model information is presented. Experiments are conducted for both the reference transcriptions and speech recogniser outputs. For reference transcription, prosodic information is shown to be more useful than language model information. A few straightforward modifications of a conventional speech recogniser allow the system to produce punctuation and speech recognition hypotheses simultaneously. The multiple hypotheses are produced by the automatic speech recogniser and are re-scored by prosodic information. When prosodic information is incorporated, the F-measure can be improved and small reductions in word error rate are obtained at the same time. An alternative approach for generating punctuation marks from the 1-best speech recogniser output which does not have any punctuation marks is also proposed. Its results are compared with those from the combined punctuation generation and speech recognition system.

Two different systems are proposed for the task of capitalisation generation. The first system is a slightly modified speech recogniser. In this system, every word in its vocabulary is duplicated: it is given once in a decapitalised form and again in a capitalised form. In addition, the language model is re-trained on mixed case texts. The other system is based on NE recognition and punctuation generation, since most capitalised words are first words in sentences or NE words. Both systems are compared first on the condition that every procedure is fully automated. The system based on NE recognition and punctuation generation shows better results in word error rate, in F-measure and in SER than the system modified from the speech recogniser. This is because the latter system has distortion of the LM, a sparser LM, and loss of half scores. The performance of the system based on NE recognition and punctuation generation is investigated by including one or more of the following: reference word sequences, reference NE classes and reference punctuation marks. The results show that this system is robust to NE recognition errors. Although most punctuation generation errors cause errors in this capitalisation generation system, the number of errors caused in capitalisation generation does not exceed the number of errors in punctuation generation. In addition, the results demonstrate that the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation.

Declaration

This thesis is the result of my own work, and where it draws on the work of others, this is acknowledged at the appropriate points in the text. Some of the work has already been, or will shortly be, published in conference proceedings [55, 57] or a technical report [56]. The length of this dissertation, including appendices and footnotes, is approximately 43,000 words.

Acknowledgements

I wish to express my gratitude to my supervisor, Phil Woodland, for his sincere leadership and encouragement. His experienced advice has guided this research and given me great inspiration.

I am also very grateful to Gavin Smith. Gavin started his Ph.D. course at the same time as me and gave me some insight into English language, culture and politics together with his brother, Nathan. I am also grateful to K. K. Chin and Matt Stuttle for creating a comfortable atmosphere. I thank other colleagues in the lab for their encouragement and discussion. In particular, Andy Tuerk and Gareth Moore helped me with parts of experimental set-ups and Sue Johnson (now Sue Tranter) gave me a brilliant introduction to the lab when I first arrived. Thanks to Patrick Gosling, who provided the superb computing environment.

The research would not have been possible without the financial support from LG company, the British Council, the Cambridge University Overseas Trust and GCHQ. Additional funding from Cambridge University Engineering Department and Darwin College enabled me to attend several international conferences. The support from all of these organisations is gratefully acknowledged.

I would like to express my thanks to my friends, who have encouraged me since I was in Seoul Science High School and KAIST. I also would like to thank Cathy Taylor, who conscientiously revised my English in this thesis.

However, above all, I appreciate my family's endless love and my parents' sacrifice. Also, I thank my maternal grandmother, who took care of me when I was a child. I am grateful too for the endless concern of two elder brothers and my cousin, Yoon-Young. I would like to recompense their favours with this thesis.

Notation

w	A word
W	A word sequence
f	The word feature of a word
c	The capitalisation type of a word
t	The Named Entity (NE) class of a word
b	The NE boundary information of a word
	If the word is combined with its previous word into a single NE word, $b = 1$ and if not, $b = 0$
z	The part-of-speech (POS) tag of a word
y	A punctuation mark
Y	A punctuation mark sequence
r	The prosody feature set for a word
R	A prosody feature set sequence
α	The scale factor for a prosodic feature model when combined with a language model

Table of contents

1 Introduction **9**
1.1 Named Entity recognition and speech recognition output enhancement 10
1.2 Key issues of the tasks 11
1.2.1 Difficulties when using speech input 12
1.3 Scope of the thesis 14
1.3.1 Named Entity (NE) recognition 14
1.3.2 Generation of punctuation 14
1.3.3 Generation of capitalisation 15
1.4 Organisation of the thesis 16

2 Previous work **17**
2.1 Named Entity (NE) recognition 17
2.1.1 Stochastic system 19
2.1.2 Rule-based system 22
2.2 Speech recognition output enhancement 26
2.2.1 Automatic punctuation generation 26
2.2.2 Automatic capitalisation generation 29
2.3 Summary 31

3 Corpora and evaluation measures **32**
3.1 Experimental data preparation 32
3.1.1 Data preparation for the development of NE recognition system 35
3.1.2 Data preparation for the development of LM 36
3.1.3 Data preparation for the development of prosodic feature model 37
3.1.4 Data preparation for the development of capitalisation generation system 38
3.2 Evaluation measures 40
3.2.1 Scoring program 42
3.3 Summary 43

4 Rule-based Named Entity (NE) recognition **44**
4.1 Transformation-based rule inference approach 44
4.2 Transformation-based automatic NE rule generation 47
4.2.1 Preprocessing 47
4.2.2 Rule-generation and testing 51
4.3 Experiments 55
4.3.1 Experimental results 55

4.3.2 Effects of punctuation and capitalisation 56
4.3.3 Effects of name lists 56
4.3.4 Effects of speech recognition errors 58
4.4 Summary 61

5 Automatic punctuation generation 62
5.1 Punctuation generation 62
5.1.1 Automatic punctuation generation for reference transcriptions 63
5.1.2 Automatic punctuation generation combined with speech recognition 65
5.2 Experiments 67
5.2.1 Classification tree setup 67
5.2.2 Results: Automatic punctuation generation for reference transcriptions 72
5.2.3 Results: Automatic punctuation generation combined with speech recognition 75
5.3 Error analysis 80
5.3.1 The effectiveness of the assumption for punctuation mark pronunciation 81
5.3.2 The effectiveness of the prosodic feature model 81
5.3.3 Estimation: Result of the punctuation generation system with speech recognition 82
5.3.4 Comparison with the system which does not use the assumption for the pronunciation of punctuation marks 86
5.3.5 The variations of punctuation marks between annotators 87
5.4 Summary 89

6 Automatic capitalisation generation 90
6.1 Capitalisation generation 90
6.1.1 Automatic capitalisation generation by modifications of speech recogniser 92
6.1.2 Automatic capitalisation generation based on NE recognition and punctuation generation 94
6.2 Experiments 97
6.2.1 Results: The system modified from a speech recogniser 98
6.2.2 Results: System based on NE recognition and punctuation generation 99
6.3 Analysis of performance of the system based on NE recognition and punctuation generation 101
6.3.1 The contribution of each experimental step 101
6.3.2 The effect of NE recognition errors 104
6.3.3 The effect of punctuation generation errors 105
6.3.4 The correlation between the effects of NE recognition errors and the effects of punctuation generation errors 106
6.3.5 Comparison with Microsoft Word 2000 107

6.3.6 Estimation: Results of the system based on NE recognition and punctuation generation when every procedure is fully automated 107
6.4 Summary 109

7 Conclusions and further work 110
7.1 Review of the contributions of this thesis 110
7.1.1 Rule-based Named Entity (NE) recognition 110
7.1.2 Automatic punctuation generation 111
7.1.3 Automatic capitalisation generation 111
7.2 Suggested further work 112
7.2.1 The use of syntactic information 112
7.2.2 More precise definition of pronunciation for punctuation marks 113
7.2.3 New NE task definition 113

Appendix 114

References 120

Chapter 1
Introduction

Considerable progress has been made in speech recognition technology over the last few decades. Recently, interest in speech recognition research has shifted from read speech data to speech data found in the real world such as broadcast news and conversational speech over the telephone. This shift opens up many applications such as information extraction systems.

Information extraction systems analyse unrestricted text in order to extract specific types of information. When searching for information of specific interest in non-textual data, such as video or audio recordings, it would be extremely useful to devise some method of automatically deriving some textual tokens from the non-textual data which would then be used to represent the content, especially when the collection is relatively large, or new items are added frequently.

These reasons have motivated the speech and computational linguistics communities to attempt to perform shallow understanding of speech beyond simply its transcription. This requires a range of techniques, including the ability to identify Named Entities (NE) - the who, where, when and how much in a sentence.

The current state-of-art technologies of speech recognition focus on producing the exact sequence of pronounced words. The readability of speech recognition output would be greatly enhanced by generating proper punctuation and capitalisation, because standard transcriptions of speech lack most capitalisation and punctuation. In addition, the generated punctuation and capitalisation give further clues for the NE recognition.

The work in this thesis concerns Named Entity (NE) recognition from speech and its application to the generation of enhanced speech recognition output including automatic punctuation and automatic capitalisation. In this introduction, first, the task of Named Entity recognition is defined, and the need for the enhancement of speech recognition output described. Then, the key issues of the tasks - especially when input comes from speech - are explained. The final section outlines the scope of the remainder of this thesis.

1.1 Named Entity recognition and speech recognition output enhancement

The NE task requires the recognition of named entities (names of locations, persons and organisations), temporal expressions (dates and times) and numerical expressions (monetary amounts and percentages) [10]. The task is to identify all instances of the three types of expression in each text in the test set, to sub-categorise the expressions, and to produce a single, unambiguous output for any relevant string in the text. An example is given in Figure 1.1[1].

> Mr <ENAMEX TYPE="PERSON"> Mandelson </ENAMEX> had made clear for the first time that all the new institutions, including the various cross-border bodies created <TIMEX TYPE="DATE"> yesterday </TIMEX> under the <ENAMEX TYPE="ORGANIZATION"> North South Ministerial Council </ENAMEX>, would all be wound up unless devolution was matched by <ENAMEX TYPE="ORGANIZATION"> IRA </ENAMEX> decommissioning.

Figure 1.1 Example of NE recognition output file

When speech dictation is performed, the dictation system can rely on the speakers to say "capitalise the current word" or "full stop" whenever they are necessary in the dictated text. However, when speakers are not aware that their speech is being automatically transcribed as in speech data found in real world (i.e. broadcast news and conversational speech over the telephone), verbalised punctuation and capitalisation are not present. Automatic punctuation and capitalisation generation will greatly enhance the readability of transcriptions, because standard transcriptions of speech lack most capitalisation and punctuation.

> Mixed case+punctuation marks+figures: One new security assessment listed the IRA as possessing at least 1,000 rifles, 500 handguns, 50 heavy machine guns and 2,600 kgs of Semtex high explosive.
>
> SNOR: ONE NEW SECURITY ASSESSMENT LISTED THE IRA AS POSSESSING AT LEAST ONE THOUSAND RIFLES FIVE HUNDRED HANDGUNS FIFTY HEAVY MACHINE GUNS AND TWO THOUSAND AND SIX HUNDRED KILO GRAMS OF SEMTEX HIGH EXPLOSIVE

Figure 1.2 Lack of capitalisation and punctuation in speech recogniser output. Speech recogniser output is conventionally written in the format of SNOR (Standard Normalised Orthographical Representation)

[1] Each NE is surrounded by its appropriate tags. 8 possible NE classes and their starting and end tags are listed in Table 3.4

As illustrated in Figure 1.2, even with no speech recognition errors, automatically transcribed speech is much harder to read due to the lack of punctuation, capitalisation and number formatting. The format of standard recogniser output, as shown in the lower part of Table 1.2, is known as Standard Normalised Orthographical Representation (SNOR) [1] and consists of only upper-case letters without punctuation marks or numbers.

The tasks of NE recognition and of enhanced speech recognition output generation are substantially related to each other, because most capitalised words apart from first words in sentences are NEs. NE recognition experiments, which compare the effects of the input condition of between mixed cases and SNOR, showed that the performance deteriorates when the capitalisation and punctuation information are missing [58]. This missing information makes certain decisions regarding proper names more difficult.

1.2 Key issues of the tasks

Although these tasks seem clear, the correct answer is not apparent in some cases due to the ambiguity in natural language. For NE recognition, ambiguous examples are discussed in [58] as follows:

- When is the Wall Street Journal an artifact, and when is it an organisation?
- When is the White House an organisation, and when is it a location?
- Are branch offices of a bank an organisation?
- Should yesterday and last Tuesday be labelled dates?
- Is mid-morning a time?

The system must produce a single, unambiguous output for any relevant string in the text. In order to encourage consistency and reduce ambiguity regarding NE recognition, guidelines have been defined in [31].

For punctuation generation, word sequences provide information about the possible locations and types of punctuation marks, but this are not sufficient. The following example, mentioned in [29], shows how different the meaning can be according to the punctuation even if the word sequence apart from punctuation is the same:

- Woman! Without her, man is nothing.
- Woman without her man, is nothing.

Many commercial implementations of automatic capitalisation generation are provided with word processors. In these implementations, grammar and spelling checkers of word processors generate suggestions about capitalisation. A typical example is one of the most popular word processors, Microsoft Word. A simple experiment was conducted using Microsoft Word 2000 for an ambiguous word, 'bill' (which can be used as a person's name as well as a statement of account). The phrase "President Bill Clinton says" was typed in de-capitalised form into Microsoft Word 2000, and only suggestions regarding capitalisations were accepted. The result was "President bill Clinton says". This example shows that capitalisation generation requires a process of dis-ambiguation of ambiguous words.

When the input text comes from speech, the NE and the speech recognition output enhancement tasks become more difficult because of corruptions in input text caused by speech recognition errors. Details are given in the following section.

1.2.1 Difficulties when using speech input

Training patterns for NE recognition, punctuation generation and capitalisation generation are designed to account for the variety of syntactic and semantic structures. Thus, patterns with several required elements are quite sensitive to errors in the input text. If any of the required elements are missing in the input, or if an extra token intervenes between the elements in the input, then the input will no longer match the pattern. An example text corrupted by speech recognition errors is shown in Figure 1.3. The example speech recognition output is taken from the output of the SRI's speech recognition system for the test data of 1998 NIST Hub-4 broadcast news benchmark test [1].

THE GUARDIANS OF THE ELECTRONIC STOCK MARKET THE NASDAQ WHO'VE BEEN BURNED BY PAST ETHICS QUESTIONS ARE MOVING TO HEAD OFF THE MARKET FRAUD BY TOUGHENING THE RULES FOR COMPANIES BUT ONE OF THE LISTED ON THE EXCHANGE MARKET PLACE IS FULL BORE OFFER FOR ITS PART OF THE PROPOSALS PENNY STOCK ALL THE ELIMINATE THE STAFF

which is a transcription of

THE GUARDIANS OF THE ELECTRONIC STOCK MARKET NASDAQ WHO'VE BEEN BURNED BY PAST ETHICS QUESTIONS ARE MOVING TO HEAD OFF MARKET FRAUD BY TOUGHENING THE RULES FOR COMPANIES THAT WANT TO BE LISTED ON THE EXCHANGE MARKETPLACE'S PHILIP BOROFF REPORTS AS PART OF THE PROPOSALS PENNY STOCKS WILL BE ELIMINATED FROM NASDAQ

Figure 1.3 Corruption in input text caused by speech recognition error. The speech recognition output is produced by the SRI system of [1].

An experiment regarding the effect of corruption caused by speech recognition errors was conducted for NE recognition in [58]. According to this experiment, NE recognition performance is sensitive to speech recognition performance, and the performance degrades linearly with increasing word error rate. An analysis of the errors made with speech recognition input showed that the dominant error was with missing names; the second most prominent error was with spurious names.

Speech disfluencies such as filled pauses and repetitions are prevalent in spontaneous speech. They are the characteristics which distinguish spontaneous speech from planned or read speech. Unlike the corruption of input which is mentioned in the previous section, these kinds of error do not come from speech recogniser errors but from the disfluencies themselves. In these cases of disfluency, any missing elements or extra intervening tokens can cause mismatches between trained patterns and input speech. Speech disfluencies can be classified based on how the actual utterance must be modified to obtain the intended fluent utterance. The classes can be characterised by the type of editing required. Their classifications are as follows, where errors are marked by an asterisk following the disfluency.

- Filled pauses

  e.g. CAMBRIDGE UH * UNIVERSITY

- Repetitions

  JOHNSON * JOHNSON WAS HERE

- Repairs

  JOHNSON * JACKSON LIKED IT

In the filled pause case, instead of recognising “CAMBRIDGE UNIVERSITY” as an organisation, “CAMBRIDGE” will be tagged as a location. In the second example, there is confusion as to whether the organisation “JOHNSON & JOHNSON” is intended, or whether the speaker accidentally repeats the name. A similar problem occurs with the third example.

1.3 Scope of the thesis

The work in this thesis concerns NE recognition from speech and its use in the generation of enhanced speech recognition output with automatic punctuation and automatic capitalisation. An automatic rule generating method is proposed for NE recognition. Punctuation marks are generated using context and prosody information. Capitalisation is produced based on the results of NE recognition and punctuation generation.

1.3.1 Named Entity (NE) recognition

In this thesis, NE recognition uses the Hub-4 IE-NE Task Definition Version 4.8 [33] as defined for the 1998 NIST Hub-4 Information Extraction (Named Entity) Broadcast News Benchmark Test Evaluation [1]. According to this definition, the NE task requires the recognition of the following NE classes:

- Named Entity: PERSON, ORGANIZATION, LOCATION
- Time expressions: DATE, TIME
- Numerical expressions: MONEY, PERCENT

Previous work regarding the NE task are mainly categorised by hand crafted rule-based systems and stochastic systems. In Chapter 4, an automatic rule generating method, which uses the Brill rule inference approach, is proposed. The performance of the rule-based Named Entity recogniser is compared with that of BBN's commercial implementation called IdentiFinder.

When only the sequences of words are available, both systems show almost equal performance as is also the case with additional information such as punctuation, capitalisation and name lists. In cases where input texts are corrupted by speech recognition errors, the performance of both systems are degraded by almost the same level. Although the rule-based approach is different from the widely used stochastic method, these results show that automatic rule inference is a viable alternative to the stochastic approach to NE recognition, while retaining the advantages of a rule-based approach.

1.3.2 Generation of punctuation

Among the many kinds of punctuation marks, this thesis is restricted to the examination of full stops, commas and question marks only. This is because there is sufficient occurrence of these punctuation marks in training corpora to obtain reliable patterns and parameters.

A punctuation generator which incorporates prosodic information along with acoustic and language model information is presented in Chapter 5. Experiments are conducted for both the reference transcriptions and speech recogniser outputs. For the reference transcriptions, prosodic information is shown to be more useful than language model information.

A few straightforward modifications of a conventional speech recogniser allow the system to produce punctuation and speech recognition hypotheses simultaneously. The multiple hypotheses are produced by the automatic speech recogniser and are re-scored by prosodic information. When prosodic information is incorporated, the F-measure can be improved and small reductions in word error rate are obtained at the same time. An alternative approach for generating punctuation marks from the 1-best speech recogniser output which does not have any punctuation mark is proposed. Its results are compared with those from the combined punctuation generation and speech recognition system.

1.3.3 Generation of capitalisation

In this thesis, capitalisation types of words are classified into three categories as shown in Table 1.1. Although there are some exceptions which do not fall into one of these three categories (e.g. McWethy, O'Brien, LeBowe), most of these exceptional words are surnames, and can be classified as Fst_Cap in Table 1.1. The details of the data preparation for capitalisation experiments are described in Chapter 3.

Capitalisation type	Description
No_Cap	Every character of a word is de-capitalised
All_Cap	All characters of a word are capitalised
Fst_Cap	Only first character of a word is capitalised

Table 1.1 Categories of capitalisation types of words

An automatic means of capitalisation is presented that uses the results of speech recognition, punctuation generation and NE recognition in Chapter 6. Experiments are conducted for both the reference transcriptions and speech recogniser outputs. Experimental results using reference transcriptions show that this automatic capitalisation method is robust to NE recognition errors and punctuation generation errors. In addition, automatic capitalisation results for speech recognition output show that this automatic capitalisation method is also robust to speech recognition errors.

1.4 Organisation of the thesis

The objective of this thesis is to devise automatic methods of NE recognition, punctuation generation and capitalisation generation from speech input. This thesis consists of seven chapters. Chapter 2 introduces previous work in this area. Chapter 3 describes the corpora used in the experiments and explains pre-processing steps used for these corpora. Also, this chapter discusses evaluation measures for the systems. Chapter 4 describes a rule-based NE recogniser. Chapter 5 presents a combined system using prosody for punctuation generation and speech recognition. Chapter 6 examines an automatic means of generating capitalisation using the NE recogniser and the punctuation generator. Finally, Chapter 7 concludes this thesis and proposes future work.

Chapter 2
Previous work

In this chapter, previous work related to NE recognition from spoken data and speech recognition output enhancement is described. Since both are relatively new areas, there are no books or journals devoted to them at this time. In Section 2.1, previous work on NE recognition is described and categorised. In Section 2.2, previous studies related to speech recognition output enhancement, mainly automatic punctuation and automatic capitalisation, are examined.

2.1 Named Entity (NE) recognition

The best source of information relating to NE recognition system descriptions is the Message Understanding Conference (MUC) Proceedings [32, 83] and the 1999 DARPA Broadcast News Workshop Proceedings [73]. These Proceedings contain the results of the performance evaluations as well as system descriptions for each participating system in the evaluation. The evaluations of MUC used domain specific text data. For MUC systems, since the domain is limited and capitalisation information helpful for detecting NEs is available, many participating systems of MUC were based on hand crafted rules. Some rule-based NE recognition systems developed for MUC-7 are described in [24, 30, 44, 90]. As the 1998 NIST Hub-4 evaluation used broadcast news data, each participant in this evaluation was required to handle various domains in broadcast news and to cope with input which does not have capitalisation information. Focusing on the 1999 DARPA Broadcast News Workshop proceedings, which contain the results of the most recent evaluation i.e. the 1998 NIST Hub-4 Information Extraction (Named Entity) Broadcast News Benchmark Test Evaluation, the general procedures used are described and previous studies are categorised. Then, each of these categories is explained.

NE recognition systems are generally categorised according to whether they are stochastic (typically HMM-based) or rule-based [56]. In the stochastic method, linguistic information is captured indirectly through large tables of statistics. However, in many instances, a stochastic system encounters difficulties in estimating probabilities from sparse training data. In contrast to the stochastic method, the rule-based method encodes linguistic information directly in a set of simple rules.

The advantages of the rule-based method over the stochastic method include its smaller storage requirements, absence of need for less-descriptive models as in *back-off* [54], and its easy extension using expert linguistic knowledge due to its conceptually reasonable rules. However, a disadvantage of previous rule-based systems is that rules need to be manually constructed [56].

Manually constructed rule-based systems show reasonable performance for normal texts because many NEs have helpful capitalisation information. However, if the input is derived from speech, capitalisation information is no longer available and it is much harder to obtain the necessary linguistic information using manually constructed rules.

In the 1998 NIST Hub-4 Information Extraction (Named Entity) Broadcast News Benchmark Test Evaluation, four sites (BBN, MITRE, SPRACH and SRI) participated and submitted their results (SPRACH implemented two systems) [1]. In this evaluation, the test data were annotated according to the Hub-4 IE-NE Task Definition Version 4.8 [33]. Table 2.1 shows the types of system as well as their performance.

Site	Type	F-measure	SER(%)
BBN	Stochastic	0.91	15.7
MITRE	Stochastic	0.88	20.3
SPRACH-R	Rule-based	0.71	46.1
SPRACH-S	Stochastic	0.83	29.1
SRI	Rule-based	0.90	16.3

Table 2.1 1998 Hub-4 NE evaluation results [1]

The BBN system is a HMM-based system known as IdentiFinder [66]. Details of stochastic NE recognition systems are described in Section 2.1.1, focusing on IdentiFinder, one of the most successful stochastic NE recognition systems. The MITRE system is another stochastic model which is similar to BBN's IdentiFinder [71]. More complete and recent descriptions of the MITRE system are given in [72]. The MITRE system uses a state topology designed for explicit modelling of variable-length phrases and class-based statistical language model smoothing. SPRACH submitted two systems: SPRACH-S [46, 78] and SPRACH-R [78]. SPRACH-S is a HMM-based system, whereas SPRACH-R is a rule based system. A standard n-gram based formulation is used in SPRACH-S. SPRACH-R uses a modified version of the NE recognition component of the Sheffield LaSIE-II system [45, 52]. Its basic approach relies on finite state matching against words stored in lists, part-of-speech tagging and phrasal grammar for the NE classes. Lastly, SRI employed TextPro which is based on the technology of the SRI FASTUS system [21]. The general processes performed by previous rule-based systems are described in Section 2.1.2. Details of finite-state cascade rule-based systems are described in Section 2.1.2.1, focusing on the FASTUS system.

2.1.1 Stochastic system

Hidden Markov models (HMMs) for NE recognition were adopted due to the success in speech recognition and have also been applied to parsing and part-of-speech tagging [34, 40]. HMMs are discussed in a number of books and tutorial papers [74, 75, 91]. In this section, details of stochastic NE recognition systems are described, focusing on IdentiFinder [23, 58, 67].

These methods regard the states of the HMM as classes of NEs. Transition probabilities are probabilities of an NE class given the previous NE class, and emission probabilities are probabilities of a word given an NE class. The probability of a particular NE class sequence given a sentence is a product of the transition and emission probabilities involved. Just as the stochastic approach to speech recognition attempts to maximise the probability of a sequence of words given a certain speech signal, the NE recogniser attempts to find the most likely sequence of NE classes given a sequence of words. Figure 2.1 shows a pictorial representation of an HMM in NE recognition.

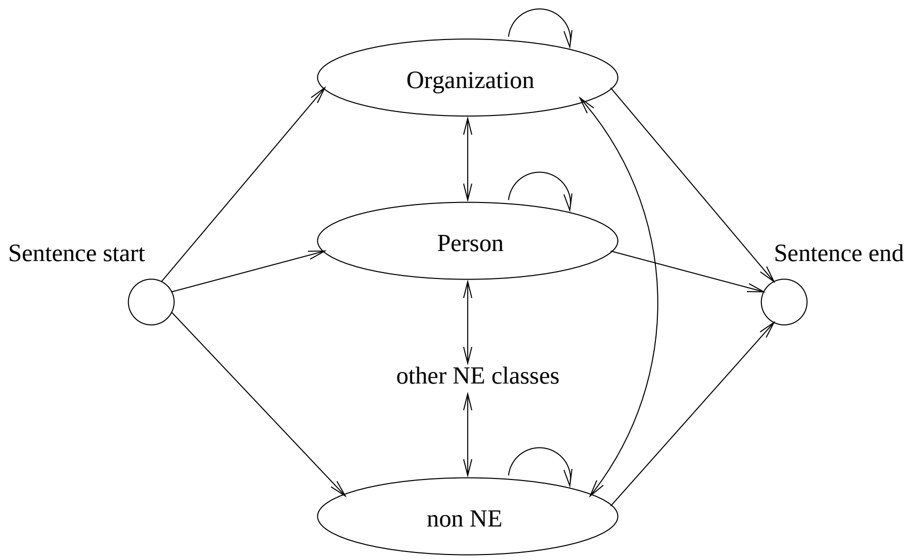

Figure 2.1 Pictorial representation of stochastic NE recogniser

Formally, we must find the most likely sequence of NE classes $t_1, ..., t_n$ given a sequence of words $w_1, ..., w_n$:

$$\max P(t_1, ..., t_n | w_1, ..., w_n) \tag{2.1}$$

Applying Bayes' rule, this can be written as:

$$\max \frac{P(t_1, ..., t_n) \times P(w_1, ..., w_n | t_1, ..., t_n)}{P(w_1, ..., w_n)} \tag{2.2}$$

The *a priori* probability of the word sequence, the denominator in equation 2.2, is constant for any given sentence. Since we are interested in finding the $t_1, ..., t_n$ that gives the maximum value in equation 2.2, the denominator in all these cases does not affect the answer. Thus, the problem reduces to finding the sequence $t_1, ..., t_n$ which maximises the following expression,

$$\max P(t_1, ..., t_n) \times P(w_1, ..., w_n | t_1, ..., t_n) = \max P(w_1, ..., w_n, t_1, ..., t_n) \tag{2.3}$$

There are still no effective methods for calculating the probability of these long sequences accurately, as it would require far too much data. But the probabilities can be approximated by probabilities that are simpler to collect, by making some independence assumptions. The probability of the sequence of NE classes can be approximated by a series of probabilities based on a limited number of previous NE classes. The most common assumptions use either one or two previous NE classes. The bigram model, using only one previous NE class, looks at pairs of NE classes and uses the conditional probability that an NE class t_i will follow an NE class t_{i-1}, written as $P(t_i|t_{i-1})$. The trigram model uses the conditional probability of one NE class given two preceding NE classes, that is, $P(t_i|t_{i-2}, t_{i-1})$.

The second probability in equation 2.3, $P(w_1, ..., w_n|t_1, ..., t_n)$, can be approximated by assuming that a word appears in an NE class independent of the words in the preceding or succeeding NE classes. It is approximated by the product of the probability that each word occurs in its indicated NE class.

$$P(w_1, ..., w_n | t_1, ..., t_n) \approx \prod_{i=1,n} P(w_i|t_i) \tag{2.4}$$

If we assume the use of bigrams, the problem changes into one of finding the sequence $t_1, ..., t_n$ which maximises the value as follows:

$$\max \prod_{i=1,n} P(w_i|t_i) \times P(t_i|t_{i-1}) \tag{2.5}$$

Using trigrams, the problem changes into

$$\max \prod_{i=1,n} P(w_i|t_i) \times P(t_i|t_{i-2}t_{i-1}) \tag{2.6}$$

Next, the most likely sequence of NE classes for a sequence of words has to be assigned. The key insight is that because of the Markov assumption, there is no need to process all the possible sequences: the assignment can be done using the Viterbi algorithm.

Due to the limited amount of training data, many of the possible bigrams will not be observed, and therefore these probabilities must be estimated using a less powerful back-off model with a suitable smoothing mechanism [18].

In many instances, a language model encounters difficulties in the estimation of probabilities from sparse training data. In the absence of further information, it seems reasonable to assume that all unseen events have equal probabilities i.e. that they are uniformly distributed. However, in language modelling, further information is often available in less-descriptive language models. For example, when using trigrams, bigrams can also be considered. This procedure of re-estimating the unseen probability using a less-descriptive model is called back-off [54].

When applying stochastic methods to NE recognition, particular importance must be given to the effect of the words which are encountered in the test data but have not been seen in the training data. For example, when using bigrams, there are three ways unknown words can appear: as current words, as previous words, or as both. One method of improvement is to build a separate unknown word model which contains statistics of unknown words. Usually, part of the training data is held out for estimating the unknown word model. For the training data which is not held out, a vocabulary list is developed. Held-out data is then analysed with the vocabulary list. Then, statistics for the occurrence of unknown words are obtained by considering the words which appear in held-out training data but not in the vocabulary list.

2.1.2 Rule-based system

In stochastic methods for NE recognition, linguistic information is only captured indirectly through large tables of statistics. Therefore, the stochastic methods need a large amount of training data in order to capture linguistic information. In the rule-based methods, linguistic information is encoded directly in a set of simple rules, in contrast to the many thousands of probabilities learned by the stochastic method. Therefore, an advantage of the rule-based system over the stochastic system is that significantly less storage is needed for pattern action rules than for an HMM-based system's probability matrix. Generally, compactness is an advantage for the rule-based system. Another general advantage is speed. Unlike stochastic systems, most rule-based systems are deterministic [92].

In NE recognition, the temporal and numeric expressions have a fairly structured appearance which can be captured by means of grammatical rules. However, person's names, organisation names and location names are more complex and more context dependent.

Rule-based NE recognition systems presented in [7, 8, 11, 12], in general, perform initial phrasing and apply hand-crafted phrase-finding rules. In this section, the general processes performed by previous rule-based systems are described with examples. Finite cascade rule-based systems are explained in the following sub-section, focusing on the FASTUS system.

In preprocessing, a set of initial phrasing functions is applied to all of the sentences to be analysed. This process is driven by word lists and part-of-speech information. Initial phrasing produces a number of phrase structures, many of which have the initial null labelling (none), while some have been assigned an initial label (e.g. number). This is done both by matching the input against pre-stored lists of proper names, date forms, currency names, etc. and by matching against lists of common nouns that act as reliable indicators or signalling words for classes of NE. An example of a set of initial phrasing functions is:

- Organisation names
- Person names
- Location names: names of major cities in the world as well as province/state and country names.
- Time expressions: phrases like 'first quarter of'
- Signalling words
  - Titles: e.g. 'President', 'Mr.'
  - Company Designator: e.g. 'Co.', 'Ltd', 'PLC'
  - Currency units: e.g. 'dollars', 'pounds'
  - Location: e.g. 'Gulf', 'Mountain'

- Organisation: e.g. 'Agency', 'Ministry' for governmental institution, 'Airline', 'Association' for companies.

Once the preprocessing has taken place, proper phrase identification proceeds. This is driven by a sequence of phrase-finding rules. Each rule in the sequence is applied in turn against all of the phrases in all of the sentences under analysis. The action can either change the label of the satisfying phrase, expand its boundaries, or create new phrases. After the nth rule has been applied in this way against every phrase in all of the sentences, the $n+1$th rule is then applied in the same way, until all the rules have been applied. Here are some examples of the named organisation grammar rule:

(Organisation Name) $\rightarrow$ (Organisation Name) (Organisation signalling word)

e.g. HMV headquarters

(Organisation Name) $\rightarrow$ (Country Name) (Content word)* (Organisation signalling word)

e.g. U.S. embassy

(Organisation Name) $\rightarrow$ (Person Name) (Content word)* (Organisation signalling word)
e.g. Lee's foundation

(Organisation Name) $\rightarrow$ (Location Name) (Content word)* (Organisation signalling word)

e.g. U.S. Defence Department

The rule (Organisation Name) $\rightarrow$ (Names) & (Names) means that if a proper name (Names) is followed by '&' and another proper name, then it is an organisation name. An example of this is "Ammirati & Puris", which matches this pattern and is therefore classified as an organisation. Rules for monetary and time expressions have been collected by analysing actual expressions in the training texts such as:

(Money expression) $\rightarrow$ (Country name)* (Number) (Money unit)

e.g. U.S. five dollars

(Money expression) $\rightarrow$ (Number) (Word)* (Country name) (Money unit)

e.g. five new Taiwan dollars, three thousand Korean Won

Rule-based systems, trained on a corpus, were developed for the MITRE system in MUC-6 [15] and for the LTG system in MUC-7 [64]. In [15], the MITRE system for MUC-6 used Brill's rule inference approach [26], but the details of how this approach were applied to the NE recognition task were not given. The LTG system for MUC-7 used probabilistic partial matching, in addition to grammars and name list look-up [64, 65]. An unsupervised algorithm using parsing results for NE recognition was described in [38], in which NE rules are generated using a parser and 7 simple seed rules.

2.1.2.1 Finite-state cascade based system

The idea of using cascaded finite state machines was pursued for POS tagging and partial parsing in [17, 42]. A finite state cascade consists of a sequence of strata, each stratum being defined by a set of regular expression patterns for recognising phrases.

Consider that a stratum has the patterns A $\rightarrow$ ab*, B $\rightarrow$ ac*. Patterns are translated by standard techniques into finite state automata. The union of all automata at a given stratum yields a single automaton. This can be done by adding arcs that output A and B, leading to new final states that have no outgoing arcs.

Adding ϵ-transitions from the new final states back to the initial state, we can make an automaton that can recognise patterns A and B repeatedly. Figure 2.2 shows such an automaton. Using ϵ-Closure, this model can be changed into a nondeterministic finite automaton [92].

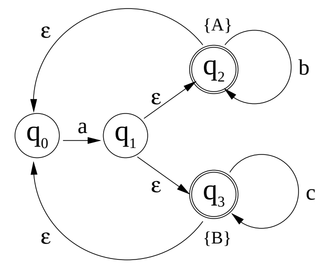

Figure 2.2 Finite state automaton accepting A $\rightarrow$ ab*, B $\rightarrow$ ac* repeatedly

For example, running this automaton against the input *abbac* produces (as one alternative) the state sequence and output shown in Figure 2.3. Multiple strata can be cascaded by using the output of a stratum as the input of the next stratum. Figure 2.4 shows the results from the two strata after adding a second stratum with pattern $C \rightarrow AB$.

Output(Stratum 1)						A				B
State (Stratum 1)	q_0	q_1	q_2	q_2	q_2	q_0	q_1	q_3	q_3	q_0
Input		a		b	b		a		c	

Figure 2.3 Results at stratum 1

Output(Stratum 2)										C
State (Stratum 2)	q_0	q_0	q_0	q_0	q_0	q_1	q_1	q_1	q_1	q_2
Output(Stratum 1)						A				B
State (Stratum 1)	q_0	q_1	q_2	q_2	q_2	q_0	q_1	q_3	q_3	q_0
Input		a		b	b		a		c	

Figure 2.4 Results at stratum 2

FASTUS (Finite State Automaton Text Understanding System) is a system for information extraction [19, 20]. In FASTUS, sentences are processed by a cascaded, nondeterministic finite-state automaton. The output of each stratum becomes the input to the next stratum. Each stratum produces some new linguistic structure, and discards some information that is irrelevant to the information extraction task. Since the automaton is nondeterministic and may produce more than one alternative, these alternatives should be compared and the best analysis selected for processing at the subsequent higher level.

FASTUS consists of five levels; preprocessor, phrase parser, phrase combiner, domain pattern recogniser and merger. The first to the third levels (the preprocessor, the phrase parser and the phrase combiner) are relevant to the NE task. The two remaining levels (the domain pattern recogniser and merger), however, produce higher level natural language processing structures, and so are not mentioned in this section. The following describes the processing stage for NE recognition in FASTUS.

1. Preprocessor:

   Names and other fixed form expressions are recognised in this stage. Complex words are recognised such as multi-words (e.g. New Taiwan Dollars) and some company names (e.g. Bridge Sports Co.). The names of people and locations, dates, times, and other basic entities are also recognised at this level.

2. Phrase parser:

   In this stage, sentences are segmented into noun groups (the part of the noun phrase consisting of determiner, prenominal modifiers and head noun), verb groups (auxiliaries, intervening adverb, and main verb), and particles (single lexical items, including conjunctions, prepositions and relative pronouns).

3. Phrase combiner:

   In this stage, complex noun groups are recognised on the basis of syntactic information. Certain prepositional phrases are attached to their noun groups, and conjunctions of noun groups are combined. This includes the attachment of “of” and “for” prepositional phrases to their head noun groups. Also, in this stage, noun groups are combined with appositives, genitives, and prepositions to provide further information about the entity (e.g. John Smith, President and CEO of Foobarco). Furthermore, adjacent location noun groups are merged (e.g. Palo Alto, California).

2.2 Speech recognition output enhancement

In this section, previous studies related to speech recognition output enhancement are examined. As standard transcriptions of speech lack most capitalisation and punctuation, the previous studies are described for the area of automatic punctuation generation and automatic capitalisation generation.

2.2.1 Automatic punctuation generation

Automatic punctuation from speech is a crucial step in making the transition from speech recognition to speech understanding. Also, automatic punctuation can greatly improve the readability of speech recognition output. The occurrences of each punctuation mark were counted in [22] for the 42 million token Wall Street Journal corpus. This study reported that about 10.5% of tokens are punctuation marks. More details are shown in Table 2.2.

Punctuation mark	Relative occurrence
,	4.658%
.	4.174%
“ ”	1.398%
( )	0.211%
?	0.039%
!	0.005%

Table 2.2 Statistics for punctuation marks in Wall Street Journal corpus [22]

2.2.1.1 Punctuation generation system using only lexical information

An automatic punctuation system, called Cyberpunc, which is based on only lexical information, was developed in [22]. Their system only produced commas, under the assumption that sentence boundaries are pre-determined. A post-processing step added commas to each punctuation-free sentence by applying an extended language model which accounts for punctuation. For a sentence which consists of n words, there are $n-1$ possible positions of commas. Among 2^{n-1} possible hypotheses containing words and commas, the best hypothesis was generated using Viterbi decoding. They claimed that this idea can be applied to the re-scoring of speech recognition lattices in general, but it was tested for a reference text (2317 reference sentences of the Penn Treebank corpus [61]) after the stripping of all punctuation marks. About 66% of commas in the reference were correctly restored, and about 76% of total generated commas in the hypothesis were correctly produced.

2.2.1.2 Punctuation generation system using acoustic and lexical information for read speech

A method of speech recognition with punctuation generation based on acoustic and lexical information was proposed in [29]. When punctuation generation is performed simultaneously with speech recognition, it is important to assign acoustic pronunciations to each punctuation mark. Punctuation marks were treated as words, and acoustic baseforms of silence, breath, and other non-speech sounds were assigned to punctuation marks in the pronunciation dictionary. A preliminary experiment was conducted for read speech. This preliminary experiment showed that only 6.5% of punctuation marks are not related to pauses and 75.6% of pauses are related to punctuation marks. Based on this result that pauses are closely related to punctuation in read speech, a speech recognition and automatic punctuation experiment was performed for 330 word business letters. Each letter was read aloud by 3 speakers. This experiment was carried out to determine how well pauses match with punctuation marks (not for punctuation recognition), using an acoustic model trained on speech from 1,800 speakers and using a language model trained on 250 million words.

2.2.1.3 Sentence boundary recogniser using lexical information and pause duration

Since many full stops and question marks are located at the end of a sentence, it is very important in punctuation generation to recognise sentence boundaries correctly. A sentence boundary recogniser using lexical information and pause duration was developed in [47]. In their work, a sentence boundary class for a word was assigned according to whether a sentence break was attached to the end of a word. Therefore, each word was assigned to either a "last-word" class or a "not-last-word" class. A sentence boundary recognition test was then developed to find the sequence of sentence boundary classes of words in speech recognition output by combining probabilities from a language model and from a pause duration model. In this work, the language model estimates the joint probability of the current word and sentence boundary class conditioned on the previous words and classes. The pause duration model can be combined with the language model based on two assumptions: first, that the previous pause duration does not affect the current word, the current sentence boundary class or the current pause duration and secondly, that current pause duration is independent of previous words and sentence boundary classes. A sentence boundary recognition experiment was conducted for 16 hours of broadcast news data using acoustic and duration models trained on 300 hours of acoustic data and using a language model trained on a 9 million words. The Word Error Rate (WER) was measured as 26.3% for the test data. This study found that a pause duration model when used alone performs better than a language model, and that the result can be improved by combining these two information sources. About 62% of sentence boundaries in reference were restored correctly, and about 80% of total generated sentence boundaries in the hypothesis were correctly produced.

2.2.1.4 Combination methodology with a language model and a prosodic feature model

It is known that there is a strong correspondence between discourse structure and prosodic information [80]. A comparison between syntactic and prosodic phrasing was presented in [43]. In his study, syntactic structures were generated by Abney's chunk parser [16] and prosodic structures were given by ToBI ([81]) label files. This work showed that at least 65% of syntactic boundaries are coded in the prosodic boundaries for read speech.

A combination methodology of intonation and dialogue context to reduce WER in speech recognition for spontaneous dialogue was described in [85]. In their research, a separate intonation model for each Dialogue Act (DA or classification whether an utterance is a statement, question, agreement and etc.) was applied to give a set of likelihoods for an utterance being one or another type of DA. Then a separate language model for each DA was applied to find the most likely DA sequence and the new speech recognition result.

In order to use prosodic information in discourse structure analysis including automatic punctuation, great attention has to be paid to how to obtain prosodic features computationally, how to build a prosodic feature model, and how to combine a prosodic feature model with models for other information sources.

A combination methodology with a language model and a prosodic feature model was discussed in [80]. In this work, the combination methodology was applied to the DA classification. For the prosodic feature model construction, 58 computable prosodic features were used. All of these features were related to duration, F0, pause, energy or speaking rate. A Classification And Regression Tree (CART) [25] was used to construct a prosodic feature model. In order to make the computation tractable, an assumption was introduced that the prosodic features were independent of the word once conditioned on the DA (a similar assumption was introduced in [85]). Experiments were performed for a 29,000 word length part of the Switchboard corpus. Experiments showed that performance was improved over that of the language model alone by integrating the prosodic model with the language model. The importance of each prosodic feature was measured by "feature usage", which is proportional to the number of times a feature was queried. According to this measure, features used higher in the tree had greater usage values than those lower in the tree. The measure "feature usage" was normalised to add up to 1.0 for each tree. In their study, duration related features were used in more than half of the queries for DA classification.

A prosodic feature model based on CART was also applied to topic segmentation in [51]. In that paper, the identification of intonational phrase boundaries using a set of acoustic features was performed using CART.

2.2.2 Automatic capitalisation generation

Another important aspect of speech recognition output enhancement is automatic capitalisation because capitalisation information also does not exist in speech input. The importance of NE recognition in automatic capitalisation was mentioned in [48]. In that study of NE tagged language models, it was stated that automatic capitalisation can possibly be achieved by programming the speech recognition decoder to produce lowercase characters apart from the capitalisation of the detected NEs. However, this is not enough for automatic capitalisation because capitalised words can normally be categorised into two groups: first words in sentences and NE words. Furthermore, some NE words are not capitalised and some non NE words are capitalised. In addition, in some capitalised words, all characters are capitalised. Therefore, systems of automatic capitalisation have to rely on NE recognition, automatic punctuation, and the capitalisation look-up table.

An approach to the disambiguation of capitalised words was presented in [63]. The capitalised words which were located at positions where capitalisation was expected (e.g. the first word in a sentence) may be proper names or just capitalised forms of common words. The main strategy of this approach was to scan the whole of the document in order to find the unambiguous usages of words.

Table 2.3 shows the statistics of 3 hours of test data from the NIST 1998 Hub-4 broadcast news benchmark test. In this database, 15.26% of total words are capitalised. As the average number of words in a sentence is 16.87, 5.23% of total words are first words in sentences. 80.45% of NE words are capitalised. Among non NE words which are not first words in sentences, 2.32% of words are capitalised.

Type	Number of occurrences
Words (any type)	31,595
Capitalised words	4,822
NE words	3,149
De-capitalised NE words	615
Capitalised non-NE words (not first word in sentence)	606
Single letter initial words (NE)	543
Single letter initial words (non-NE)	78
Sentences	1,873

Table 2.3 Number of occurrences of different word capitalisations in the NIST 1998 Hub-4 broadcast news test data

2.2.2.1 Grammar and spelling checker in Microsoft Word

Many commercial implementations of automatic capitalisation are provided with word processors. In these implementations, the grammar and spelling checkers of word processors generate suggestions about capitalisation. A typical example is one of the most popular word processors, Microsoft Word. The details of its implementation was described in a U.S. patent [77]. In this implementation, whether the current word is at the start of a sentence was determined by a sentence capitalisation state machine. A word was defined as the text characters and any adjacent punctuation. The sentence capitalisation state machine used the characters of the current word for the transition between its possible states. For example, if it passes a sentence ending punctuation character, the capitalisation state machine changed its state to the end punctuation state. By passing the characters of words to the capitalisation state machine, the auto correct function could determine if a particular word is at the end of a sentence, and if so, the auto correct function could determine that the next word needs to begin with an upper case letter. The capitalisation of words which are not the first words in sentences could be found by dictionary look-up. When a word was entered in all lower case, the capitalisation was applied for the word to have the greatest consistency in matching the capitalisation.

When input comes from speech, automatic capitalisation becomes more difficult because sentence boundary information and capitalisation information are not available in natural speech. For example, a broadcast news transcription system cannot rely on the speakers to say “capitalise the current word” or “full stop” whenever they are necessary in the transcribed text. Reliable results for automatic capitalisation can be obtained for speech input by using the results of NE recognition in conjunction with automatic punctuation. As both NE recognition and automatic punctuation are relatively new areas, it is currently difficult to find papers related to automatic capitalisation for speech input.

2.3 Summary

This chapter has described work in the field of NE recognition, automatic punctuation and automatic capitalisation. NE recognition systems are generally categorised according to whether they are stochastic or rule-based. The advantages of the rule-based NE recognition system over the stochastic method include the fact that there is no need for less-descriptive models as in back-off due to its conceptually reasonable rules. However, the rule-based system has disadvantages in portability if its rules are manually constructed.

Automatic punctuation is a relatively new research area. Previous work reported very promising results, but they are limited in the use of information sources, experimental assumptions and the domain of test data. Other related work highlights the possibility of performance improvements in automatic punctuation through the combination of prosodic features with other information sources.

Many commercial implementations of automatic capitalisation are provided with word processors. These implementations are based on sentence boundary detection and dictionary look-up. However, dictionary look-up is not enough for dis-ambiguation of words which can be used in both the de-capitalised and the capitalised forms. In addition, sentence boundary information does not exist if input comes from speech.

This survey suggests that reliable results of automatic capitalisation may be obtained for speech input by using the results of NE recognition in conjunction with automatic punctuation.

Chapter 3
Corpora and evaluation measures

This chapter begins with descriptions of the use of corpora and preprocessing used in language model construction, Named Entity recognition, punctuation generation and capitalisation generation. It goes on to describe the scoring metrics and the scoring program used in this thesis.

3.1 Experimental data preparation

Language models, NE recognisers, punctuation generation systems and capitalisation generation systems derive their parameters and patterns from a large text corpus and a large amount of acoustic training data. Two different sets of data, the Broadcast News (BN) text corpus and the 100-hour Hub-4 BN data set, are available as training data for the experiments conducted in this thesis. The BN text corpus (named BNtext92_97 in this thesis) comprises a 184 million word BN text over the period of 1992-1997 inclusive[1]. Another set of training data, the 100-hour BN acoustic training data set released for the 1998 Hub-4 evaluation (named DB98) consists of acoustic data and its detailed transcription.

Broadcast News provides a good test-bed for speech recognition, because it requires systems to handle a wide range of speakers, a large vocabulary, and various domains. Three hours of test data from the NIST 1998 Hub-4 broadcast news benchmark tests are used as test data for the evaluation of the proposed systems. This test data is named TDB98. TDB98 comprises 3 hours of acoustic data and the transcription. Table 3.1 summarises the training and test data.

Name	Description	#Words	Purpose	Acoustic data
BNtext92_97	1992_97 BN texts	184M	Training data	Not available
DB98	100 hrs of Hub-4 data (1998)	774K	Training data	Available
TDB98	1998 benchmark test data	32K	Test data	Available

Table 3.1 Experimental data descriptions

[1]The 1992-1996 part is provided by the LDC and the 1997 part is provided by the Primary Source Media.

The BN transcriptions are used to capture the sequence of spoken words. They may also include annotations which associate speaker, signal and recording conditions. In DB98 and TDB98, the sequence of words is enclosed by corresponding tags which identify the location of the speech within the speech signal using start and end time tags. Also, NE words are enclosed by NE tags in DB98 and TDB98. An example of the data is shown in Table 3.2.

<Turn startTime=“4052.108937” endTime=“4064.492000” spkrtype=“male” dialect= “native” speaker=“Craig_Wintom” mode=“planned” fidelity=“high”>
More snow is falling this morning in northern <b_enamex TYPE=“LOCATION”> Ohio <e_enamex> and other parts of the <b_enamex TYPE=“LOCATION”> Great Lakes <e_enamex> region,
<time sec=“4057.162187”>
tens of thousands of homes remain without electricity.
<time sec=“4060.432187”>
From member station <b_enamex TYPE=“ORGANIZATION”> _W_C_P_N <e_enamex> in <b_enamex TYPE=“LOCATION”> Cleveland <e_enamex>, <b_enamex TYPE=“PERSON”> Joe Smith <e_enamex> reports.
</Turn>

Table 3.2 Example data file

As different data source uses different tags, headings, and punctuation mark definition, preprocessing steps are necessary to ensure compatibility with other data. In addition, it is necessary to keep compatibility with the vocabulary of the speech recogniser, because NE recognition, punctuation generation and capitalisation generation will be carried out on speech recognition output. The following steps are applied to training and test data:

- Headings: Headings are removed from transcriptions, because in general they are not grammatically correct.
- Tags: Tags are discarded, but NE start tags and NE end tags are treated differently from other tags to keep NE information.
- Punctuation: Punctuation marks are written as special words (e.g. ,COMMA) in some parts of the data and they are attached to the previous word in the other part. As punctuation marks are used by language models, punctuation marks are separated from the previous word and are written as special words.
- Genitive: Genitive forms such as ’s and ’ are separated from their previous words by NE tags when the previous words are transcribed as NE words. For example, “Mr. <b_enamex TYPE=”PERSON”>Clinton<e_enamex>’s past”. During an NE recognition, every genitive word is separated and dealt with as a separate word. After NE recognition finishes, these

genitive words are attached to their previous words and the NE class of the previous words are maintained.

- Abbreviation: A period is attached to its previous word in some parts of the data, but an underscore is attached instead (e.g. _C_N_N) in other parts of the data. In order to keep consistency with the vocabulary of the speech recogniser, underscores are replaced by periods and abbreviated words are separated.
- De-hyphenation: Hyphenated words are separated to reduce the Out-Of-Vocabulary (OOV) rate since many hyphenated pairs may not appear in the vocabulary of the speech recogniser while the constituent words do appear.
- Noises: Noise markers such as "{LAUGH", "{BREATH" and "{LIPSMACK" are removed.

Training data are used for three different tasks: NE recognition, automatic punctuation generation and automatic capitalisation generation. As described in Section 1.3, these three tasks require the development of an NE recogniser, a Language Model (LM) which includes punctuation marks, a prosodic feature model, and a capitalisation generator.

Each set of training data has different characteristics and information. In addition, acoustic data is not available for BNtext92_97 while it is available for DB98. Regarding the development of an NE recogniser, only the transcription of DB98 was used as training data because BNtext92_97 does not contain NE tags. Both BNtext92_97 and DB98 can be used for the LM development. As this LM is used within the speech recogniser, the transcriptions of BNtext92_97 and DB98 are converted into single-case retaining punctuation marks to produce LM probabilities for punctuation marks. However, only DB98 is used for the implementation of a prosodic feature model, because acoustic data are not available for BNtext92_97.

Although both BNtext92_97 and DB98 are case-sensitive, the consistency of capitalisation is poor in BNtext92_97. Sometimes, all characters of a sentence are capitalised in BNtext92_97, but it is impossible to remove these words in the preprocessing steps because they are not contained by tags. For this reason, only the transcription of DB98 is used as the training data for the capitalisation process.

Developed system (or model)	BNtext92_97	DB98
NE recognition	Not used	Used
LM (punctuation inclusive)	Used	Used
Prosodic feature model	Not used	Used
Capitalisation generation	Not used	Used

Table 3.3 Usage of training data for each system development

Table 3.3 summarises the training data used for the system developments of NE recognition, LM, prosodic feature model, and capitalisation generation. The statistics of the data for each development, and the necessary preparations for them, will be presented in the following sections.

3.1.1 Data preparation for the development of NE recognition system

NE tags were annotated for DB98. This data is available from the LDC (LDC98E11 [3]). The DB98 is used as training data for the development of an NE recognition system in this thesis. The DB98 is provided in the Universal Transcription Format (UTF) format: documentation with more information on the annotation is available in [13]. Each NE in the training data and the output produced by NE recognition systems, should be surrounded by its appropriate tags. Table 3.4 lists the 8 possible NE classes used in this task and their starting and end tags.

NE class	Starting tag	End tag
ORGANIZATION	<b_enamex TYPE="ORGANIZATION">	<e_enamex>
PERSON	<b_enamex TYPE="PERSON">	<e_enamex>
LOCATION	<b_enamex TYPE="LOCATION">	<e_enamex>
DATE	<b_timex TYPE="DATE">	<e_timex>
TIME	<b_timex TYPE="TIME">	<e_timex>
MONEY	<b_numex TYPE="MONEY">	<e_numex>
PERCENT	<b_numex TYPE="PERCENT">	<e_numex>
non-NE	Nothing	Nothing

Table 3.4 Possible NE classes and their surrounding tags

In the NIST 1998 Hub-4 broadcast news benchmark test, MITRE and SAIC provided 3 hours of test data. It contains 1,765 tagged entities. The ENAMEX tag is the dominant entity type and represents 88% of all tagged entities in the test data whereas both the TIMEX and the NUMEX entities represent only 6% of the entities in the test data [73]. Because the test data adopts the same annotation as DB98, and because it is easy to compare performance to other systems which participated in the NIST 1998 benchmark test, the 3 hour test data is used as test data for NE recognition experiments in this thesis. Tables 3.5 and 3.6 show the statistics of the training and test data for the development of the NE recognition system.

Name	Usage	Number of words	Vocabulary size
DB98	Training data	773,893	28,344
TDB98	Test data	31,595	5,429

Table 3.5 Statistics of data in the development of NE recognition system

	Number of tagged entities		Number of tagged words	
NE class	DB98	TDB98	DB98	TDB98
ORGANIZATION	9,033	415	21,215	953
PERSON	13,427	436	20,833	717
LOCATION	12,139	714	16,556	934
MONEY	1,162	79	3,951	275
PERCENT	643	25	1,666	89
DATE	2,766	80	5,151	137
TIME	275	16	858	44
Total	39,445	1,765	70,230	3,149

Table 3.6 Statistics of NE classes in the development of NE recognition system

3.1.2 Data preparation for the development of LM

An LM was developed to obtain the LM probabilities of hypothesis which includes punctuation marks. In this thesis, the HTK BN transcription system is used in the generation of punctuation marks. More details about the development of the HTK BN transcription system are given in [89].

Punctuation marks are retained in both BNtext92_97 and DB98. A trigram and a 4-gram LM were developed on these data to produce hypotheses which contain punctuation marks and to expand the generated hypotheses. As the HTK BN transcription system produces single-case speech recognition outputs, the transcriptions of BNtext92_97 and DB98 are converted into single-case. Among the many kinds of punctuation marks, this thesis is restricted to the examination of full stops, commas, and questions marks, because there are sufficient occurrences of these punctuation marks in the training and test corpora.

When automatic punctuation is simultaneously performed with speech recognition, it is important to assign acoustic pronunciations to each punctuation mark. The correlation between punctuation and pauses was investigated in [29]. These experiments showed that pauses closely correspond to punctuation marks. The correlation between pause lengths and sentence boundary marks was studied for broadcast news data in [47]. In that study, it was observed that the longer the pause duration, the greater the chance of a sentence boundary existing. Although some instances of punctuation do not occur at pauses, it is convenient to assume that the acoustic pronunciation of punctuation is silence. Full stops, commas, and questions marks are included in the 108K size vocabulary of the HTK BN transcription system and their pronunciation is given as silence in the pronunciation dictionary. Table 3.7 shows the statistics of the training and test data for the development of LM.

Name	Number of occurrences			
	Words	Commas	Full stops	Question marks
BNtext92_97	184M	11.7M	10.9M	1.3M
DB98	774K	30,063	42,609	2,470
TDB98	32K	1,491	1,653	101

Table 3.7 Statistics of data for the development of LM

3.1.3 Data preparation for the development of prosodic feature model

Many easily computable prosodic features were investigated for Dialog Act (DA) classification in [80]. In their study, 58 computable prosodic features were used for the prosodic feature model construction. All of these features were related to duration, F0, pause, energy or speaking rate. A Classification And Regression Tree (CART) [25] was used to construct a prosodic feature model.

In this thesis, a set of 10 prosodic features is investigated for punctuation generation through a consideration of the automatic punctuation task and the contribution of each prosodic feature for DA classification. The end of each word is a possible candidate for punctuation, and so all prosodic features are measured at the end of a word. The window length is set at 0.2 secs. The left window is the window to the left of the word end, and the right window to the right. Good F0 values are those greater than the minimum F0 (50Hz) and less than the maximum F0 (400Hz). Table 3.8 explains these features.

Name	Description
Pau_Len	Pause length at the end of a word
Dur_fr_Pau	Duration from the previous pause
Avg_F0_L	Mean of good F0s in left window
Avg_F0_R	Mean of good F0s in right window
Avg_F0_Ratio	Avg_F0_R/Avg_F0_L
Cnt_F0_L	No. of good F0s in left window
Cnt_F0_R	No. of good F0s in right window
Eng_L	RMS energy in left window
Eng_R	RMS energy in right window
Eng_Ratio	Eng_R/Eng_L

Table 3.8 Description of the prosodic feature set used for the development of prosodic feature model (Window length = 0.2 sec, 50Hz $\leq$ good F0 $\leq$ 400Hz)

As speech signals are available for DB98 and TDB98, a time-alignment process can be performed between the raw speech signals and transcriptions. After obtaining the alignment results, prosodic features are extracted at the end of each word.

3.1.4 Data preparation for the development of capitalisation generation system

Automatic capitalisation generation requires case-sensitive transcriptions as its training data. Both BNtext92_97 and DB98 are case-sensitive, but consistency in capitalisation is not maintained for the whole of BNtext92_97. Sometimes, all characters of a sentence are capitalised in BNtext92_97. However, it is impossible to remove these words in the preprocessing steps, since these words are not contained by tags. For this reason, only DB98 is used as the training data in this study for the development of the capitalisation generation system.

As DB98 and TDB98 were transcribed for the speech recognition task, there are many errors in the transcription of capitalisation information. In TDB98, 97 words which are the first words in sentences are not capitalised. In addition, 14 words after commas are capitalised. These errors were corrected manually. Consistency of capitalisation were not kept between the same words in similar contexts for 79 cases. These cases were also manually corrected. This manual adjustment process is carried out throughout TDB98. Fragments and backchannels (e.g. uhhuh) are adjusted, if adjustments were necessary. As the number of words in DB98 is more than 700,000, this manual adjustment is not performed for DB98.

Capitalisation types are categorised as to whether all of the characters in a word are capitalised or de-capitalised, or whether only the first character of a word is capitalised. Details of these categories are described in Table 3.9. Capitalised length-one words such as initials in B. B. C. are categorised as All_Cap. In DB98 and TDB98, there are 437 (0.05% of total words in DB98) and

Type	Description
No_Cap	Every character is de-capitalised
All_Cap	All characters are capitalised
Fst_Cap	Only first character is capitalised

Table 3.9 Possible capitalisation type

26 exceptional cases respectively which are not categorised as any of the categories in Table 3.9. Most of these are surnames. For example, McWethy, MacLaine, O'Brien, LeBowe and JonBenet. All of these exceptional cases were checked manually. From this investigation, it was concluded that there is no exceptional case which cannot be treated as Fst_Cap. All of these exceptional cases were therefore classified as Fst_Cap. Table 3.10 shows the number of occurrences for each type of word based on the position of words in a sentence. Table 3.11 shows the statistics of data for the development of the capitalisation generation system.

Word type		#FW		#non FW	
NE class	Capitalisation type	DB98	TDB98	DB98	TDB98
NE	No_Cap	16	0	12,110	615
NE	All_Cap	536	20	10,535	577
NE	Fst_Cap	3,529	143	43,459	1,790
non NE	No_Cap	1,587	24	638,477	26,134
non NE	All_Cap	2,842	83	6,887	141
non NE	Fst_Cap	37,659	1,603	16,256	465

Table 3.10 Number of occurrences of different types of capitalisation for each type of words (FW: a first word in a sentence, non FW: not a first word in a sentence)

Type	Number of occurrences	
	DB98	TDB98
Words (any type)	773,893	31,595
Capitalised words	121,703	4,822
NE words	70,230	3,149
Single letter initial words (NE)	10,200	543
Single letter initial words (non-NE)	2,099	78
Sentences	46,169	1,873

Table 3.11 Statistics of data for the development of the capitalisation generation system

3.2 Evaluation measures

Evaluation of a system involves scoring the automatically annotated hypothesis text against a hand annotated reference text. Scoring of a text input is relatively simple because it compares expressions in the reference to those in the hypothesis text and counts the number of expressions which match in terms of type and boundary.

However, when the input comes from speech, because of recogniser deletion, insertion and substitution errors, a straightforward comparison is no longer possible [49]. Instead, the reference and hypothesis texts must first be automatically aligned. This is a complex process and involves attempting to determine which part of recogniser output corresponds to which part of the transcript.

Once the alignment is completed, correct/incorrect decisions for all the slots can be made. Define the following symbols:

$$\begin{aligned}
C &= \text{number of correct slots}\\
S &= \text{number of substitution errors}\\
D &= \text{number of deletion errors}\\
I &= \text{number of insertion errors}\\
N &= \text{number of slots in reference}\\
M &= \text{number of slots in hypothesis}
\end{aligned}$$

From the above definitions, it is clear that:

$$\begin{aligned}
\mathrm{N} &= \mathrm{C} + \mathrm{S} + \mathrm{D}\\
\mathrm{M} &= \mathrm{C} + \mathrm{S} + \mathrm{I}
\end{aligned}$$

Two important metrics for assessing the performance of an information extraction system are *recall* and *precision*. These terms are borrowed from the information retrieval community. Recall (R) refers to how much of the information that should have been extracted was actually correctly extracted. Precision (P) refers to the reliability of the information extracted. These quantities are defined as:

$$P = \frac{\text{number of correct slots}}{\text{number of slots in hypothesis}} = \frac{C}{M} \tag{3.1}$$

and

$$R = \frac{\text{number of correct slots}}{\text{number of slots in reference}} = \frac{C}{N} \tag{3.2}$$

Although theoretically independent, in practice recall and precision tend to operate in trade-off relationships. When you try to increase recall, you often lose precision. When you optimise precision, you do so at the cost of recall.

The F-measure [60] is the uniformly weighted harmonic mean of precision and recall:

$$F = \frac{RP}{(R+P)/2} = \frac{2C}{N+M} \tag{3.3}$$

Another evaluation metric called Slot Error Rate (SER) was defined in [60] as follows:

$$\text{SER} = \frac{\text{number of slot errors}}{\text{number of slots in reference}} = \frac{S+D+I}{N} \tag{3.4}$$

The difference between SER and $(1-F)$ is the weight given to each type of error. $(1-F)$ is calculated as:

$$(1-F) = \frac{N+M-2C}{N+M} = \frac{S+(D+I)/2}{(N+M)/2} \tag{3.5}$$

In $(1-F)$, deletion and insertion errors are de-weighted. It was reported in [60] that the SER is about 50% higher than the $(1-F)$ for the best performing system in the MUC-6 test.

In NE recognition, a correct slot is one in which the NE class and both boundaries are correct. A slot is half correct if the NE class is correct and the string in the slot overlaps with the reference string. Alternatively, a slot is half correct if the type of the NE class (rather than the NE class) and both boundaries are correct. The types of NE classes are defined as follows:

- Entity: PERSON, ORGANIZATION, LOCATION
- Time expressions: DATE, TIME
- Numerical expressions: MONEY, PERCENT

The same ideas of precision, recall, F-measure and SER can also be applied to punctuation and capitalisation generation. In these cases, a slot is half correct if the position of the slot is correct, but the type of the slot is generated as another type.

3.2.1 Scoring program

The NE recognition systems are evaluated based on how their output compares with the manually annotated output. The Message Understanding Conference (MUC) community has worked for several years with NE recognition for newswire text. However, newswire text assumes no speech recognition errors in the hypothesis files. Therefore, the need to allow for speech recognition errors arises. NIST worked with SAIC to develop scoring software for the task, which involved the creation of a Recognition and Extraction Evaluation Pipeline (REEP) to combine the NIST transcription filtering and SCLITE scoring software with the MUC scorer [2, 6].

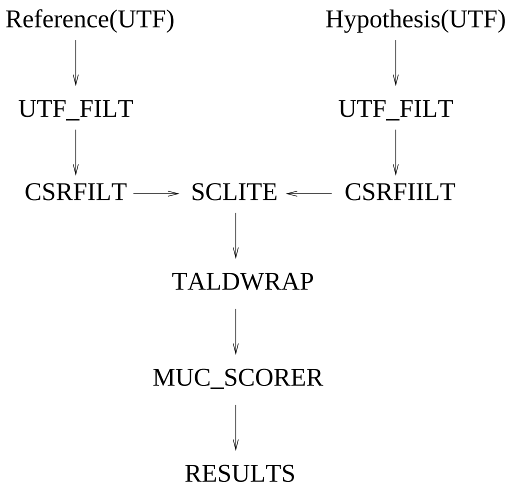

Figure 3.1 Procedures in the scoring pipeline [2]

When the scorer is run, it reads a reference file and a hypothesis file produced by the NE recogniser. The scorer aligns words in the reference file with words in the hypothesis file. It then calculates scores based on how well the entities in the hypothesis file agree with those in the reference file. In this thesis, version 0.7 of the NIST Hub-4 IE scoring pipeline package [5] is used. Figure 3.1 shows the procedures in the scoring pipeline.

Although this scoring pipeline was developed for the NE recognition system evaluation only, this scoring pipeline can be applied for the evaluation of a capitalisation generation system by small manipulations of the reference and the hypothesis files.[2] According to the definition of half scoring in the evaluation of an NE recognition system, a half score is given when the position of capitalisation is correct, but the type of capitalisation is recognised as the other type. The same manipulation tactic can be applied for the evaluation of a punctuation generation system.

[2] Surround the words whose capitalisation types are All_Cap by the “ORGANIZATION” NE class starting and end tags and enclose the words whose types are Fst_Cap by the “PERSON” NE class tags.

3.3 Summary

In this chapter, the experimental data have been described and the preprocessing which is necessary in order to use this data has been explained. The characteristics of the data have been presented for each task: the development of an NE recognition system, an LM, a prosodic feature model, and a capitalisation generation system. The F-measure, SER, precision, and recall have been described as the evaluation metrics used in this thesis. The NIST Hub-4 IE scoring pipeline package has been described, which is used as the evaluation program later in this thesis.

Chapter 4
Rule-based Named Entity (NE) recognition

In this chapter, a rule-based (transformation-based) NE recognition system is proposed. This system uses the Brill rule inference approach. The performance of the rule-based system and IdentiFinder are compared. In the baseline case (no punctuation and no capitalisation), both systems show almost equal performance.

They also have similar performance in the case of additional information such as punctuation, capitalisation and name lists. The performance of both systems degrade linearly with the number of speech recognition errors, and their rates of degradation are almost equal. These results show that automatic rule inference is a viable alternative to the HMM-based approach to NE recognition, but it retains the advantages of a rule-based approach.

In Section 4.1, Brill's transformation-based rule inference approach is introduced. In Section 4.2, a transformation-based rule-based system which generates rules automatically is presented. Then, in Section 4.3, experiments and their results are described. Finally, this chapter is summarised in Section 4.4.

4.1 Transformation-based rule inference approach

Unlike the stochastic method, one problem with the traditional rule-based method is that a large amount of effort is required to write the rules [23]. In addition to being difficult to create manually, the resulting processing systems are expensive to port to new languages or even to new domains. It is very difficult to manually encode all of the information necessary to make a robust system.

A system that automatically extracts linguistic generalisation from a corpus has two strong advantages. First, the total development time can be greatly reduced. Secondly, a system based on the analysis of a corpus can avoid over-generalisation because it learns the statistical properties [26].

Brill developed a rule based part-of-speech (POS) tagger which acquires rules from corpora [26, 27, 28]. In his work, the learning procedure begins by using an unannotated input text. At each stage of learning, the learner finds the transformation rules which when applied to the corpus result in the best improvement in tagging performance. The improvement can be calculated by comparing the current tags after the rule is applied with the reference tags. This is an important difference between a stochastic method and a transformation-based method. The stochastic method attempts to maximise the probability of input[1], while the transformation-based method attempts to minimise the number of errors. After finding this rule, it is stored and applied in order to change the current tags. This procedure continues until no more transformations can be found. Figure 4.1 illustrates the learning process.

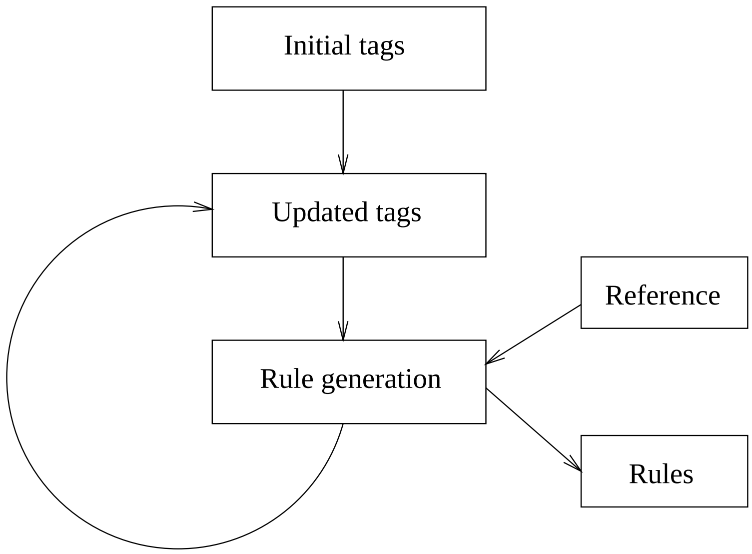

Figure 4.1 Transformation-based error driven learning

In order to define a specific application of the transformation-based method, the following must be specified:

1. The initial annotator (preprocessing)
2. The rule generation engine which examines each transformation
3. The scoring function for comparing the current tags with the reference and choosing the best transformation

Tagging accuracy was used as the scoring function in Brill's research.

[1]Maximum Likelihood (ML) training is assumed.

Rules are generated according to their rule templates at each iteration of the rule generation process. In the implementation of the Brill POS tagger, 21 rule templates were used [28]. The following rule templates are listed in [28]:

Change POS tag z_i at the position i to tag $z_i^{'}$ when:

- The preceding (following) word is tagged z
- The preceding (following) word is w
- The word two before (after) is w
- One of the two preceding (following) words is tagged z
- The current word is w and the preceding (following) word is $w^{'}$
- The current word is w and the preceding (following) word is tagged z

An example of rule generated for POS tagging is:

"Change the tag of a word from VERB to NOUN if the previous word is a DETERMINER".

Once an ordered list of transformation rules has been learned, new text is annotated by simply applying each transformation in order to the new text.

Brill's transformation-based POS tagger was compared to one of the most successful stochastic POS taggers in [27]. The results of the stochastic POS tagger using the Penn Treebank Tagged Wall Street Journal Corpus originated in [86]. In order to make reasonable comparisons, Brill's POS tagger was examined on the same corpus. In this comparison, the transformation-based POS tagger achieved better performance, despite the fact that the contextual information was captured in only 267 simple rules, whilst 10,000 contextual probabilities had been learned by the stochastic POS tagger.

The idea of the rule-based NE recognition system, which will be described in the following section, comes from the Brill POS tagger. Several systems use the Brill POS tagger simply as a preprocessor for their NE recognition systems [45, 53]. In the implementation of an NE recognition system, the Brill tagger is actually used for building the NE system; that is, all NE recognition rules are automatically generated using this idea.

4.2 Transformation-based automatic NE rule generation

Figure 4.2 illustrates the procedures in the proposed transformation-based rule-based system which automatically generates rules. The procedures are mainly divided into two parts; preprocessing, and automatic rule generation. The preprocessing steps will be explained in Section 4.2.1. Then the automatic rule generation steps, the general idea of which originated from Brill's POS tagger [26], will be described in Section 4.2.2.

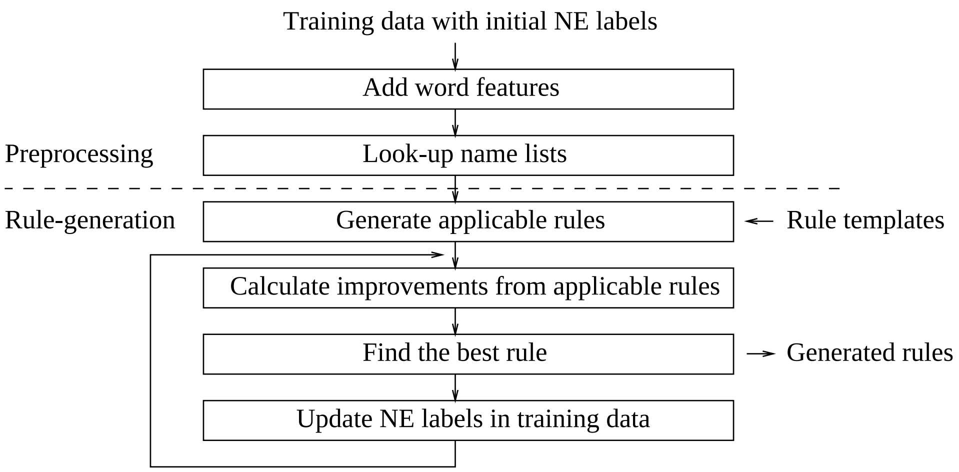

Figure 4.2 Procedures for preprocessing and rule-generation

4.2.1 Preprocessing

In this system, an untagged training data file is passed through the initial NE recogniser. It is not efficient to store words in memory and on disk as sequences of characters because of their storage requirements and the irregularity in their word lengths. Every word in the training data in this system is converted into an index in a corresponding word list in which all words are listed in their capitalised form. Indices 0, 1 and 2 are reserved for special words: sentence start (+START+), sentence end (+END+) and unknown word (+UNKNOWN+) respectively. When genitive "words" such as ' and 'S are combined with NE words, the recognition system separates these genitive words from the NE words; For example, <ENAMEX TYPE="ORGANIZATION"> NASDAQ </ENAMEX>'S. Therefore, when the system makes its word list, every genitive word is separated and dealt with as a separate word.

The syntactic structure of a sentence is in part indicated by punctuation marks, such as commas and full-stops. It is assumed in rule generation, that a sequence of words is unstructured across syntactic boundaries; but obviously this is not true [22, 29]. Therefore, if all punctuation marks are provided with the transcriptions, then the system's performance will improve. The system developed in this thesis separates all punctuation marks from consecutive words, and treats the punctuation marks as words. Figure 4.3 shows an example conversion of words to indices.

W	+Start+	Wages	in	the	United	States	have	gone	up	only	about	three
IW	0	23333	10682	21629	22700	20488	9790	9205	22844	14856	70	21748

W	and	a	half	percent	in	the	past	year	,	while	global	competition	is
IW	844	14	9593	15668	10682	21629	15472	24038	11	23657	9139	4275	11338

W	one	reason	for	the	slow	growth	in	pay	.	+End+
IW	14847	17372	8473	21629	19837	9453	10682	15552	12	1

Figure 4.3 An example of conversion from words to indices in the word list (W: Word; IW: Index of word in word list)

As some NEs consist of more than one word, it is important in the implementation of an NE recognition system to keep NE boundary information whether the word is combined with its surrounding word. For example, although the NE classes of “Tony” and “Blair” are the same,

<ENAMEX TYPE=“PERSON”> Tony </ENAMEX> <ENAMEX TYPE=“PERSON”> Blair </ENAMEX>

and

<ENAMEX TYPE=“PERSON”> Tony Blair </ENAMEX>

are different. In the implementation, storage is allocated for each word to keep the NE boundary information. Each allocated storage is set to be 0 at the initialisation. Then, if the current word is combined with the previous word into a single NE word, the value of the storage for the NE boundary information is changed to 1.

The characteristics of the word itself, called the word features, sometimes give good clues for NE recognition [23, 87]. For example, capitalisation of the first character of a word, when it is not the first word of a sentence, shows a higher possibility of being a proper noun NE word. Table 4.1 shows possible word features. First, deterministic computation is performed to obtain word features. The first two word features (Fst_Cap and All_Cap) are determined by whether the characters in these words are capitalised. The next three features (Not_in_Ent, Ent_in_L and Ent_in_R) are used to observe the relationships of non-NE words to NE words. These features can be obtained by consulting a table, which was built when the word list was made.

The last feature, NUMERIC, comes from the need to distinguish numeric and temporal entities. These features can be extracted by looking them up in a numeric dictionary, which is constructed manually. The current system uses a 63 word numeric dictionary. Since the word features are non-disjoint, one word can have more than one word feature.

Type	Descriptions
Fst_Cap	Words with capitalised first character except first words of sentences
All_Cap	Words with all capitalised characters (such as NASDAQ) and having a word length greater than 2 letters
Not_in_Ent	Words which are never used inside NEs
Ent_in_L	Words which are Not_in_Ent and which have the possibility of having an entity word on their left side
Ent_In_R	Words which are Not_in_Ent and which have the possibility of having an entity word on their right side
NUMERIC	Numeric words in the numeric dictionary

Table 4.1 Word features

A fundamental restriction of the corpus-based approach to name finding is the relatively small number of names (of people, places, organisations etc.) observed in even a large training corpus [56]. Even with the use of an unknown word model, identification of these entities depends largely upon the presence of signalling words. An extension to this approach in this system is the use of lists of location names, first names, well-known surnames, organisations etc. The advantage of this approach is that many names can be included very quickly: an enormous corpus would be necessary in order to include the same number of names from normal text.

There is generally predictive initial evidence regarding the class of a desired entity. However, it would not be desirable to decide an NE solely from its initial evidence. Consider one member of this list - “Berlin”. Although a great number of occurrences in the test data will have the location entity, we must not prevent “Berlin Orchestra” from being given the correct organisation entity. Therefore it is necessary to somehow use these lists to add information during training also. This approach is adopted in this thesis.

For lists such as first names and locations, no contextual information is available. However, in the organisation list, names usually consist of multiple words which could be used as context. In this case, the names routinely contain words such as “of” and “the”, which are entered into the rules or into the language model as occurring in the organisation entity class. Because of the large number of entries in the list, this has the effect of distorting the rules and the language model such that many occurrences of these words in the test data are mistakenly tagged when they should not be.

In this system developed here, word features from name lists can be added as word features at the preprocessing stage. Table 4.2 shows word features derived from name lists. Name lists for persons, locations and organisations are used. When the rule-based system incorporates this information, the system prefers the longer element, if more than one name-list's elements are overlapped. If the same word appears on more than one name list, then a precedence rule is applied. The location name list has the highest priority, the person name list has the next, and the organisation name list has the lowest.

Type	Description
In_P_List	Words in the persons' name list
In_L_List	Words in the locations' name list
In_O_List	Words in the organisations' name list

Table 4.2 Word features derived from name lists

Figure 4.4 summarises the results of the preprocessing stage. w_i denotes the word at the position i. In the rule generation process, which will be described in Section 4.2.2, rules are generated by comparing the NE classes and their boundaries in the current text with those in the reference. In order to perform this comparison, the NE class of w_i is kept as t_i^R in the reference. The definition of NE classes was shown in Table 3.4. In addition, b_i^R, which indicates whether w_i is combined with w_{i-1} into a single NE word (i.e. Labour Party), is also stored in the reference. If w_i is combined, b_i^R=1 and if not, b_i^R=0.

During the preprocessing stage, the initial tags are configured. f_i, which implies the word feature of w_i, is set by the characteristics of w_i and by looking-up name lists. Details of word characteristics were given in Table 4.1 and the used name lists are listed in Table 4.2.

The applicable rules are generated based on the values of w_i, f_i, t_i, and b_i to reduce the difference between t_i and b_i in the current text and t_i^R and b_i^R in reference. The initial value of t_i is configured as non-NE, and that of b_i is set to 0. Details of the generation of applicable rules will be explained in Section 4.2.2.

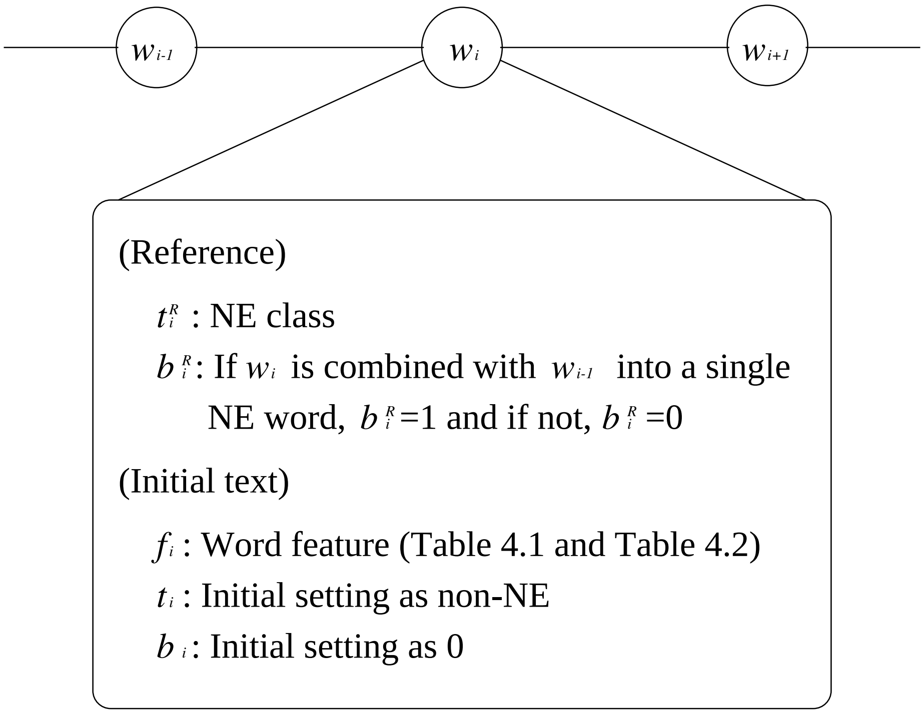

Figure 4.4 Pictorial representation of the preprocessing stage in the transformation-based automatic NE rule generation

4.2.2 Rule-generation and testing

After these preprocessing steps are completed, automatic rule-generation starts with the assignment of the NE class to every word with a non-NE tag. Once the training data file has been passed through the initial NE recogniser, its assigned NE classes and NE boundaries are compared to the true NE classes and NE boundaries, and errors are then counted. For all words whose NE classes and NE boundaries are incorrect, the rules to recognise these NE classes and NE boundaries correctly are generated and stored, and then applied, and the resulting number of improvements on the whole training data calculated. The rules are generated according to their appropriate rule templates.

Table 4.3 shows the 53 rule templates used in this system. Rule templates consist of pairs of characters and a subscript. w, f, t denotes that templates are related to words, word features and NE classes respectively. b indicates whether the word is combined with the previous word into a single NE word (if combined, b=1 and if not, b=0). Subscripts show the relative distance from the current word; that is 0 means the current word, -1 means the previous word and 1 means the next word. Rule templates have one more slot at the end. This indicates the number of the NE class of the change after the rule is activated. The definition of NE classes was shown in Table 3.4.

Stage No.	Rule+Range		
0	$w_0\ f_0$ [0 0],	$w_0\ f_{-1}$ [-1 0],	$w_0\ f_1$ [0 1]
1	$w_0\ w_1$ [0 1],	$w_0\ w_{-1}$ [-1 0],	$w_0\ t_1$ [0 1]
	$w_0\ t_{-1}$ [-1 0],	$w_1\ t_0$ [0 1],	$w_{-1}\ t_0$ [-1 0]
	$t_0\ t_1$ [0 1],	$t_0\ t_{-1}$ [-1 0],	$w_0\ f_{-1}$ [-1 0]
	$w_0\ f_1$ [0 1]		
2	$w_0\ w_{-1}\ w_{-2}$ [-2 0],	$w_0\ w_1\ w_2$ [0 2],	$w_0\ w_{-1}\ w_1$ [-1 1]
	$w_0\ t_1$ [0 1],	$w_0\ t_{-1}$ [-1 0],	$w_1\ t_0$ [0 1]
	$w_{-1}\ t_0$ [-1 0],	$w_0\ w_1\ t_2$ [0 2],	$w_0\ w_1\ t_{-1}$ [-1 1]
	$w_0\ t_1\ w_2$ [0 2]		
3	$w_0\ f_0\ b_0$ [0 0],	$w_0\ f_0\ b_0\ b_1$ [0 0]	
4	$w_0\ w_{-1}\ t_0\ t_{-1}$ [0 0],	$w_0\ w_1\ t_0\ t_1$ [0 0]	
5	$w_0\ f_0$ [0 0]		
6	$w_0\ t_0\ t_{-1}$ [-1 0],	$w_0\ t_0\ t_1$ [0 1]	
7	$w_{-1}\ w_{-2}\ t_0\ f_0$ [0 0],	$w_1\ w_2\ t_0$ [0 0],	$w_{-1}\ t_0$ [0 0]
	$w_1\ t_0$ [0 0]		
8	$w_{-1}\ f_{-1}\ f_0$ [-1 0],	$w_1\ f_1\ f_0$ [0 1],	$w_0\ f_0\ t_{-1}$ [-1 0]
	$w_0\ f_0\ t_1$ [0 1]		
9	$w_{-1}\ f_{-1}\ f_0$ [0 0],	$w_1\ f_1\ f_0$ [0 0],	$w_0\ f_0\ t_{-1}$ [0 0]
	$w_0\ f_0\ t_1$ [0 0]		
10	$w_0\ t_{-1}\ t_1\ f_0$ [-1 1],	$w_0\ f_{-1}\ f_1\ f_0$ [-1 1],	$w_0\ f_1\ w_2$ [0 0]
	$w_0\ f_{-1}\ w_{-2}$ [0 0],	$w_0\ f_1\ t_2$ [0 0],	$w_0\ f_{-1}\ t_{-2}$ [0 0]
11	$w_0\ w_{-1}$ [0 0],	$w_0\ w_1$ [0 0],	$w_0\ w_{-1}\ w_{-2}$ [0 0]
	$w_0\ w_1\ w_2$ [0 0],	$w_0\ w_{-1}\ w_1$ [0 0]	

Table 4.3 Developed rule templates (w:words; f:word features; t:NE classes). Subscripts define the distance from the current word and bracketed numbers indicate the range of rule application [start-offset from current word, end-offset from current word].

Each rule template has its own range of application where the conditions of the rule are met. For example, consider a generated rule 'if w_0 = DOLLARS and f_{-1} = NUMERIC then change NE class to MONEY'. This is for the rule template $w_0\ f_{-1}$ with range [-1 0]. This means that if the current word is 'DOLLARS' and the feature of the previous word is 'NUMERIC' then change the NE classes of the previous and current words into 'MONEY'. Then combine the previous word and the current word into a single NE word such as <NUMEX TYPE="MONEY"> five dollars </NUMEX>.

The improvement for each possible rule is updated each time a rule is generated. If all 53 rule templates are used at the same time, the computational load for this update is too heavy. In order to reduce this computational load, rule templates are grouped into 12 sets and the stages of the rule generation process are split up based on the sets of rule templates. From all the

Rule	Template
If the current word is 'DOLLARS' and the feature of the previous word is 'NUMERIC', then change the NE classes of the current and previous words to 'MONEY'	w_0 f_{-1} [-1 0]
If the current word is 'NINETEEN' and the feature of the current word is 'NUMERIC', then change the NE class of the current word to 'DATE'	w_0 f_0 [0 0]
If the current word is 'PERCENT' and the feature of the previous word is 'NUMERIC', then change the NE class of the current and previous words to 'PERCENT'	w_0 f_{-1} [-1 0]
If the current word is 'DOLLAR' and the feature of the previous word is 'NUMERIC', then change the NE classes of the current and previous words to 'MONEY'	w_0 f_{-1} [-1 0]
If the current word is 'CLINTON' and the first character of the current word is capitalised, then change the NE class of the current word to 'PERSON'	w_0 f_0 [0 0]
If the current word is 'HOUSE' and the first character of the current word is capitalised, then change the NE class of the current word to 'ORGANIZATION'	w_0 f_0 [0 0]

Table 4.4 The six rules and their rule templates which give greatest improvements at the start of training

possible rules at each stage, the rule which causes the greatest improvement is applied to the current training data and the training data file is updated. If there are any changes in NE classes or NE boundaries which affect any of the other rules, then the improvements from those other rules are also updated. In this system, the improvement is defined as the number of words which obtain their correct NE class or NE boundary after the rule is applied. These steps are repeated until no further changes can be made to the rules so as to reduce the number of errors between the current NE classes and NE boundaries for the training data and the true NE classes and NE boundaries. Table 4.4 shows the 6 rules which give greatest improvements when the training procedure starts.

In testing, the rules are applied to the input text one-by-one according to a given order. If the conditions for a rule are met, then the rule is triggered and the NE classes of the words are changed if necessary.

Particular importance must be given to the effect of words encountered in the test data which have not been seen in the training data. One way of improving the situation is to build separate rules for unknown words. The training data are divided into two groups. If words in one group are not seen in the other group, these words are regarded as unknown words. The same rule generation procedures are then applied.

Figure 4.5 summarises the procedures of transformation-based automatic NE rule generation. The complete procedure starts with the initial annotation of the text. Details of this initial annotation were given in Section 4.2.1. For all words whose NE classes and NE boundaries are incorrect, rules to recognise these NE classes and NE boundaries correctly are generated. The rules are generated according to 53 rule templates, which were listed in Table 4.3.

Among all the possible rules, the rule which reduces the errors in NE classes and NE boundaries in the current text by the greatest number is applied to the current text and which is then updated. Details of the generation of rules were given in Section 4.2.2. These steps are repeated until there is no rule which can reduce the differences. Rule are generated one-by-one. Examples of generated rules were illustrated in Table 4.4.

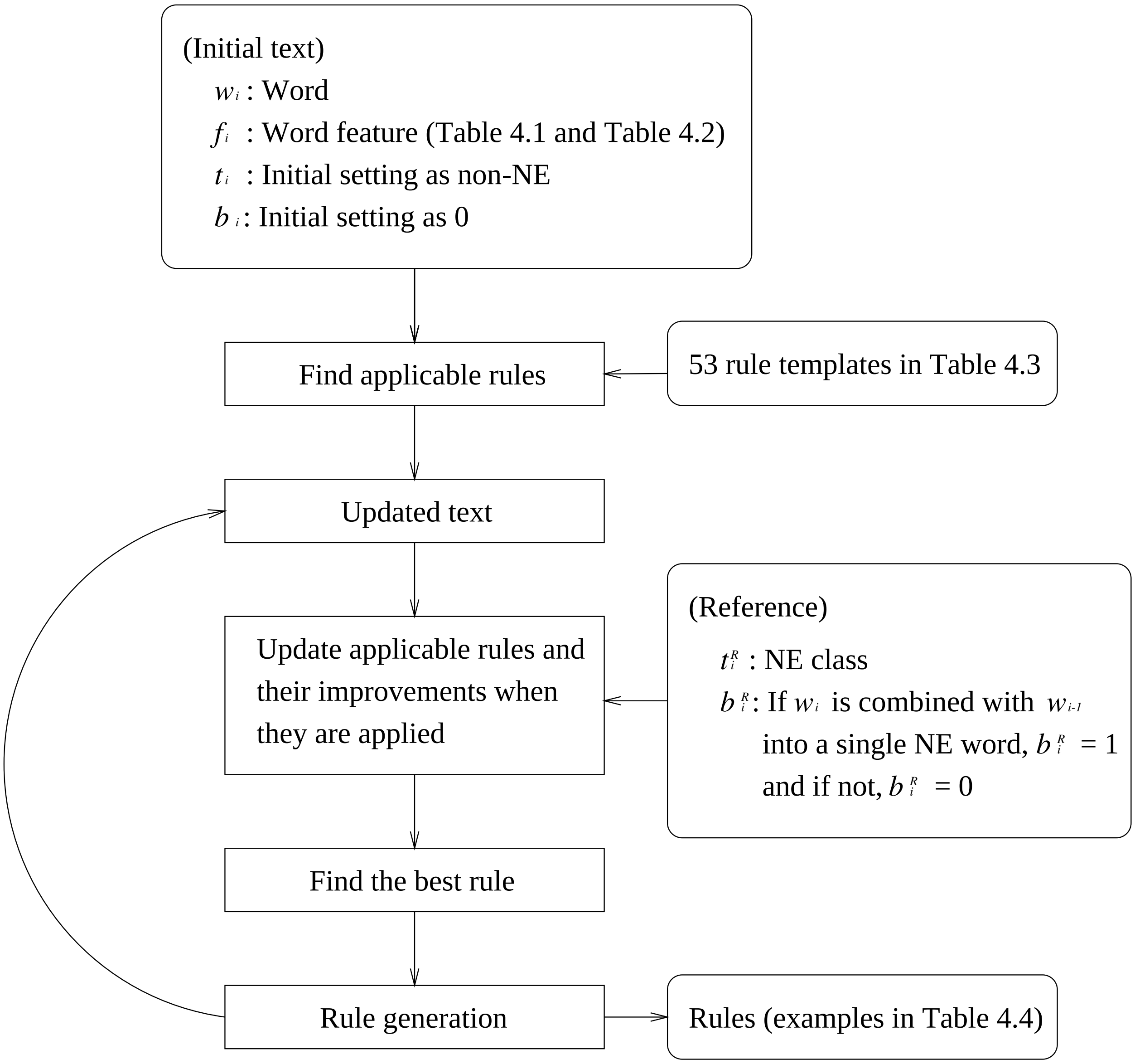

Figure 4.5 Pictorial representation of transformation-based automatic NE rule generation

4.3 Experiments

In order to measure the performance of the rule-based system, it was compared to that of IdentiFinder, BBN's HMM-based system which gave the best performance among the five systems that participated in the 1998 Hub-4 broadcast news benchmark tests [1, 73]. Compared to the results in the benchmark tests, the results of IdentiFinder shown in the following sections differ slightly, because of differences in the amount of the training data [66] and preprocessing steps for the texts. Also, there may be a difference in the version of IdentiFinder used.

In the following sections, the results of both systems are examined first in the baseline condition (with no punctuation, no capitalisation, and no name list). Then the improvement of both systems from the baseline condition is investigated for the additional textual cues of punctuation and capitalisation. In addition, the effects of name lists are discussed for both systems. Finally, degradation of performance is tested for speech recognition errors and the degree of degradation is compared for both systems.

4.3.1 Experimental results

The 100-hour 1998 Hub-4 BN data set (DB98) is used for the development of the rule-based system and IdentiFinder. These systems are evaluated in terms of F-measure and SER using the NIST Hub-4 IE scoring pipeline package for the 3 hours of data from the NIST 1998 Hub-4 BN benchmark tests (TDB98). Further details about the data, the scoring program, and the evaluation metrics were given in Chapter 3.

The performance of the rule-based system is compared with that of IdentiFinder in the baseline condition (with no punctuation, no capitalisation, and no name list). For this comparison, the training and the test data are converted into single-case and un-punctuated texts. Then, both systems are trained and tested without the use of any name lists. Table 4.5 shows the performance of each system for the baseline case. Compared to IdentiFinder, the rule-based system showed a small improvement of 0.0012 in the F-measure, but showed a small degradation of 0.35% in SER.

	F-measure		SER(%)	
Condition	RBS	IDF	RBS	IDF
Baseline	0.8858	0.8846	20.03	19.68

Table 4.5 Performance of systems for the baseline case using reference text (RBS: Rule-based system; IDF: IdentiFinder; SER: Slot Error Rate; Baseline: no punctuation, no capitalisation, and no name list)

4.3.2 Effects of punctuation and capitalisation

Next, the effect of punctuation was measured. Both systems use punctuation marks as separate "words". In order to measure how much improvement in performance is caused by the addition of this punctuation information, both systems were trained on the fully punctuated text. Punctuation has a positive effect in NE recognition and increased the performance in terms of F-measure for the rule-based system by 0.0043 and for the IdentiFinder system by 0.0074. In terms of SER, these positive effects are measured as 0.93% and 1.29% for the rule-based system and IdentiFinder respectively.

The effect of capitalisation is also measured. Capitalisation information is also used as features in both system. In order to measure how much the inclusion of capitalisation information contributes to performance, both systems are trained on the mixed case text without punctuation marks. Capitalisation information is shown to be helpful for NE recognition. In terms of F-measure, it contributes 0.0146 for the rule-based system and 0.0154 for the IdentiFinder system. In terms of SER, it contributes 3.48% and 3.08% for the rule-based system and IdentiFinder respectively.

Table 4.6 shows these results. The conditions of 'Baseline+Punctuation' are punctuation, no capitalisation and no name lists. The conditions of 'Baseline+Capitalisation' are capitalisation, no name lists and no punctuation. The addition of capitalisation information improves the performance of a system more than the addition of punctuation information.

Condition	F-measure		SER(%)	
	RBS	IDF	RBS	IDF
Baseline+Capitalisation	0.9004	0.9000	16.55	16.60
Baseline+Punctuation	0.8901	0.8920	19.10	18.39
Baseline	0.8858	0.8846	20.03	19.68

Table 4.6 Effects of punctuation and capitalisation. (SER: Slot Error Rate; RBS: Rule-based system; IDF: IdentiFinder; Baseline+Capitalisation: capitalisation, no name list and no punctuation; Baseline+Punctuation: punctuation, no name list and no capitalisation)

4.3.3 Effects of name lists

In order to investigate the effects of name lists, the rule-based NE recognition system and IdentiFinder are trained on SNOR data with name lists. Like the rule-based system, IdentiFinder can incorporate the NE information from name lists as word features, not as hard-decision rules [23]. When the rule-based system incorporates this information, the system prefers the longer element, if more than one name list's elements are overlapped. If the same word appears on more than one name list, then a precedence rule is applied. The location name list has the highest priority, the person name list has the next, and the organisation name list has the lowest.

The effects of name lists are shown in Table 4.7. The conditions of 'Baseline+NL' are with name lists, but without punctuation or capitalisation. In terms of F-measure, the use of name lists improves the performance of the rule-based system by 0.0104 and that of IdentiFinder by 0.0108. In terms of SER, it contributes 2.27% and 1.98% for the rule-based system and IdentiFinder respectively.

Condition	F-measure		SER(%)	
	RBS	IDF	RBS	IDF
Baseline+NL	0.8962	0.8952	17.76	17.70
Baseline	0.8858	0.8846	20.03	19.68

Table 4.7 Effects of name lists. Experiments were done at the baseline condition, but with name lists. (NL: Name Lists; RBS: Rule-based system; IDF: IdentiFinder; SER: Slot Error Rate)

Table 4.8 summarises the effects of capitalisation, punctuation and name lists on performance. The mixed case data with punctuation marks are processed to make four different versions: one with mixed case words and punctuation marks maintained, one with mixed case words but punctuation marks removed, one with single case words but punctuation marks maintained, and one with single case words and punctuation marks removed. For each version, both the rule-based system and IdentiFinder are trained with name lists and without name lists. The 8 different conditions reflecting these possible combinations of training and test conditions are presented in Table 4.8.

Condition	F-measure		SER(%)	
	RBS	IDF	RBS	IDF
Baseline+Cap+NL+Punc	0.9134	0.9145	13.98	14.15
Baseline+Cap+NL	0.9105	0.9121	14.72	14.30
Baseline+Cap+Punc	0.9086	0.9087	15.04	15.11
Baseline+Cap	0.9004	0.9000	16.55	16.60
Baseline+NL+Punc	0.9007	0.9010	16.68	16.69
Baseline+NL	0.8962	0.8952	17.76	17.70
Baseline+Punc	0.8901	0.8920	19.10	18.39
Baseline	0.8858	0.8846	20.03	19.68

Table 4.8 Comparison of results (Cap: Capitalisation; NL: Name Lists; Punc: Punctuation; RBS: Rule-based system; IDF: IdentiFinder; SER: Slot Error Rate)

Using the additional textual cues (punctuation and capitalisation) and name lists together, the results are improved substantially: in terms of F-measure by 0.0276 for the rule-based system and by 0.0299 for IdentiFinder. The amounts of improvement of NE recognition system from the effects of punctuation, capitalisation and name lists are measured as 0.0043, 0.0146 and 0.0104 in F-measure for the rule-based system respectively and as 0.0074, 0.0154 and 0.0106 in F-measure for IdentiFinder respectively. The improvements in both systems from adding these three additional sets of information are slightly less than the sum of individual improvements. This suggests that there are some NE words which can be corrected by additional textual cues as well as name lists. Surprisingly, for the case of "Baseline + Cap + Punc", the amount of actual improvement in both systems from the baseline condition is greater than the sum of individual improvements by capitalisation and punctuation. It is believed that there are some NE words where both mixed case and punctuation are necessary to make both systems answer correctly. The same conclusion can be drawn when the results are analysed in terms of SER.

In NE recognition, the SER is proportional to the (1.0 - F-measure). The SER is about 60% to 70% higher than the (1.0 - F-measure) in general. In Table 4.8, the rule-based system showed slightly better results in F-measure, but slightly poorer results in SER for the cases of Baseline and "Baseline+NL". For the cases of "Baseline+Cap+NL+Punc", "Baseline+Cap+Punc" and "Baseline+NL+Punc", opposite results can be observed. As explained in Section 3.2, the difference between SER and (1.0 - F-measure) is the weight to the number of each type of error. In (1.0 - F-measure), deletion and insertion errors are de-weighted.

From the results in Table 4.8, it is observed that the performances of both systems are very similar and that the amount of performance improvements from the baseline based on different conditions are almost the same. From this observation, it is concluded that both systems have almost the same ability for NE recognition. An example of NE recognition output produced for the case of "Baseline+NL+Punc" is shown in Figure 2 in the Appendix.

4.3.4 Effects of speech recognition errors

The trained patterns for NE recognition are designed to account for the variety of syntactic and semantic structures. Thus, patterns with several required elements are quite sensitive to errors in the input text: if any of the required elements are missing, or if an extra token intervenes between the elements, then the pattern will not match the input.

In order to examine the effects of speech recognition errors, experiments are conducted using the output from 11 different speech recognition systems from the 1998 Hub-4 evaluation. These outputs are available from [1]. Experiments are performed with no punctuation and no capitalisation, but still using name lists. The rule-based system and IdentiFinder are trained using the human transcribed training data.

Speech recogniser output is provided in “ctm” format [4]. Because sentence boundaries are not specified in ctm files, an alignment procedure between the reference files and speech recogniser output ctm files is needed to insert sentence boundaries. This procedure is complicated because

1) in cases where there is a speech recognition error just in front of or just next to a sentence boundary, there is uncertainty in the exact location of the sentence boundary;

2) fragments, overlapped and unclear parts in the reference are not shown in the ctm file;

3) large mismatches between a reference utf file and a ctm file are found at the locations where many speech recogniser errors occurred.

An alignment program based on the dynamic programming method was implemented to cope with these problems.

The performance of the rule-based system and IdentiFinder were evaluated on the output of the speech recognition systems for the 1998 Hub-4 evaluation. The results are presented in Table 4.10 and those in F-measure are plotted in Figure 4.6.

Although the points in Figure 4.6 are sparse, it appears that the NE recogniser performance degrades linearly with increasing Word Error Rate (WER). The line in Figure 4.6 is the line-of-best-fit for the results of the rule-based system, estimated by the least squares method [41, 62]. This line fits the data very well. For the human generated transcription, this line very slightly underestimates the result. It appears that both systems lose about 0.0062 points in F-measure per 1% of additional errors. Table 4.9 shows the decrease in F-measure for each percentage increase in WER.

System	F-measure loss
RBS	0.00627
IDF	0.00622

Table 4.9 Decrease in F-measure for each percentage increase in WER, estimated by the least squares method (RBS: Rule-based system; IDF: IdentiFinder)

The same experiment is conducted for the SER values. The SER increases linearly with increasing WER. Using the least squares method, the SER of the rule-based system is increased by 1.050% per 1% of additional WER, and the SER of the IdentiFinder by 1.043%. The two systems showed almost the same ability of labelling NE words correctly in the presence of speech recognition errors.

System	WER(%)	F-measure		SER(%)	
		RBS	IDF	RBS	IDF
human transcription	0.0	0.8962	0.8952	17.76	17.70
ibm1	13.5	0.8051	0.8018	31.48	31.71
ibm2	13.6	0.8056	0.8003	31.28	32.27
limsi1	13.6	0.8146	0.8088	29.43	30.59
cu-htk1	13.8	0.8169	0.8099	30.46	31.05
ibm3	14.1	0.8012	0.7935	33.34	35.50
dragon1	14.5	0.8053	0.8059	31.33	32.03
bbn1	14.7	0.8096	0.7999	31.30	33.33
philips rwth1	17.6	0.7888	0.7878	34.92	34.69
sprach1	20.8	0.7618	0.7611	41.23	40.30
sri1	21.1	0.7700	0.7649	38.66	39.43

Table 4.10 Effects of speech recogniser errors (WER: Word Error Rate; SER: Slot Error Rate; RBS: Rule-based system; IDF: IdentiFinder)

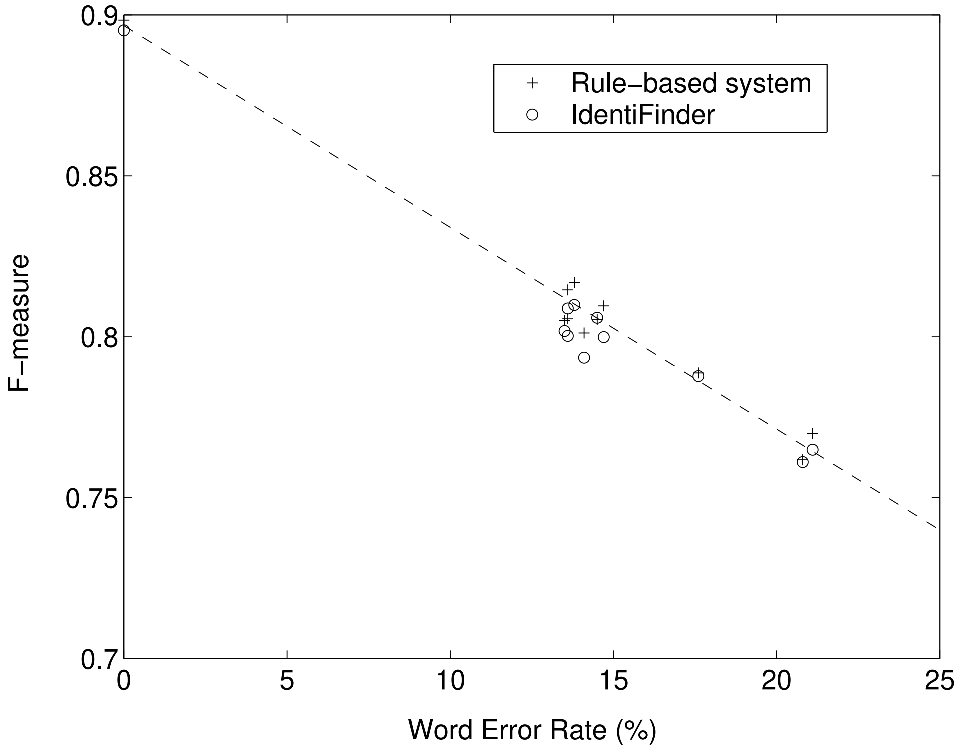

Figure 4.6 Effects of speech recogniser output errors. The line indicates the line-of-best-fit for the rule-based system's results

4.4 Summary

In this chapter, a rule-based system, which generates rules automatically, was devised. Then its performance was compared with BBN's commercial stochastic NE recogniser called IdentiFinder. For the baseline case, both systems show almost equal performance, and are also similar when additional information such as punctuation, capitalisation and name lists is given. When input texts are corrupted by speech recognition errors, the performance of both systems are degraded by almost the same amount. Although the rule-based approach is different from the stochastic method, which is recognised as one of the most successful methods, the rule-based system shows the same level of performance.

Chapter 5
Automatic punctuation generation

In this chapter, a combined system for punctuation generation and speech recognition is described. This system incorporates prosodic information with acoustic and language model information. Experiments are conducted for both the reference transcriptions and speech recogniser outputs. For the reference transcription case, prosodic information is shown to be more useful than language model information. When these information sources are combined, an F-measure of up to 0.7830 for punctuation generation can be obtained.

A few straightforward modifications of a conventional speech recogniser allow the system to produce punctuation and speech recognition hypotheses simultaneously. The multiple hypotheses are produced by the automatic speech recogniser and are re-scored by prosodic information. When prosodic information is incorporated, the F-measure can be improved by 19% relative. At the same time, small reductions in word error rate are obtained.

In Section 5.1, a methodology for automatic punctuation generation is presented. The experiments and results are then discussed in Section 5.2. The errors are analysed in Section 5.3. Finally, this chapter is concluded in Section 5.4.

5.1 Punctuation generation

In this section, a methodology for automatic punctuation generation is described for both the reference transcriptions and with speech recognition. When automatic punctuation generation is performed with the reference texts, the sequences of words are already given. Therefore, experiments aim at generating punctuation marks between words. As sentence boundary marks (<s> and </s>) provide a lot of information for locating punctuation near to them, it is unrealistic to include this information at the input for punctuation generation. Therefore, the sentence boundary marks are removed from the training and test data.

When automatic punctuation generation is performed simultaneously with speech recognition, the approximate sentence boundary marks are generated by recogniser segmentation. Sentence boundary marks are therefore not removed in this case, because the recogniser is part of the automatic punctuation generation system.

5.1.1 Automatic punctuation generation for reference transcriptions

Let Y be the punctuation mark sequence, W be the word sequence and R be the corresponding prosodic feature sequence. The automatic punctuation system aims to find the maximum *a posteriori* Y, Y_{MAP}, given W and R.

$$Y_{MAP} = \arg_Y \ \max P(Y|W,R) \tag{5.1}$$

Now

$$\begin{aligned} P(Y|W,R) &= \frac{P(Y,W,R)}{P(W,R)} && (5.2)\\ &= \frac{P(Y,W,R)\frac{P(Y,W)}{P(Y,W)}\frac{P(W)}{P(W)}}{P(W,R)} && (5.3)\\ &= \frac{\frac{P(Y,W,R)}{P(Y,W)}\frac{P(Y,W)}{P(W)}}{\frac{P(W,R)}{P(W)}} && (5.4)\\ &= \frac{P(R|Y,W)P(Y|W)}{P(R|W)} && (5.5) \end{aligned}$$

Since Y is independent of the evidence $P(R|W)$,

$$P(Y|W,R) \quad \propto \quad P(R|Y,W)P(Y|W) \tag{5.6}$$

Assuming that R depends only on Y, and $P(R)$ is uniformly distributed,

$$P(R|Y,W) = P(R|Y) = \frac{P(Y|R)P(R)}{P(Y)} \propto \frac{P(Y|R)}{P(Y)} \tag{5.7}$$

Let y_i be the ith punctuation mark and r_i be the ith prosodic feature. Apply the 1st order Markov assumption i.e.

$$p(y_i|r_1,...,r_n) = p(y_i|r_i) \tag{5.8}$$

and also let y_i be conditionally independent i.e.

$$p(y_1,...,y_n|R) = \prod_{i=1}^{n} p(y_i|R) \tag{5.9}$$

$P(Y|R)$ becomes

$$P(Y|R) = \prod_{i=1}^{n} p(y_i|r_i) \tag{5.10}$$

The probabilities in Equation 5.10 can be obtained, for instance, from the terminal nodes of classification trees (This process will be described in Section 5.2.1). $P(Y|W)$ in Equation 5.6 can be obtained from a statistical language model. $P(Y)$ can be obtained from training data counts.

The systems presented in this thesis generate only full stops, commas, and question marks. For the length n input $w_1, ..., w_n$, which does not have punctuation marks, the end of each word is a possible candidate for punctuation. Considering the three types of punctuation marks and No-Punctuation (NP), there are 4^n possible hypotheses for the input $w_1, ..., w_n$. The search for the best hypothesis can be achieved with the Viterbi search algorithm. Using this algorithm, the required time for the search for the best hypothesis is reduced to linear to the length of input n. Figure 5.1 shows a sample Viterbi search process for the generation of punctuation for an example reference transcription. The bold line in Figure 5.1 depicts the best hypothesis. For this hypothesis, commas are generated at the end of the words "pensioners" and "savers". Details of the Viterbi search algorithm are given in [76].

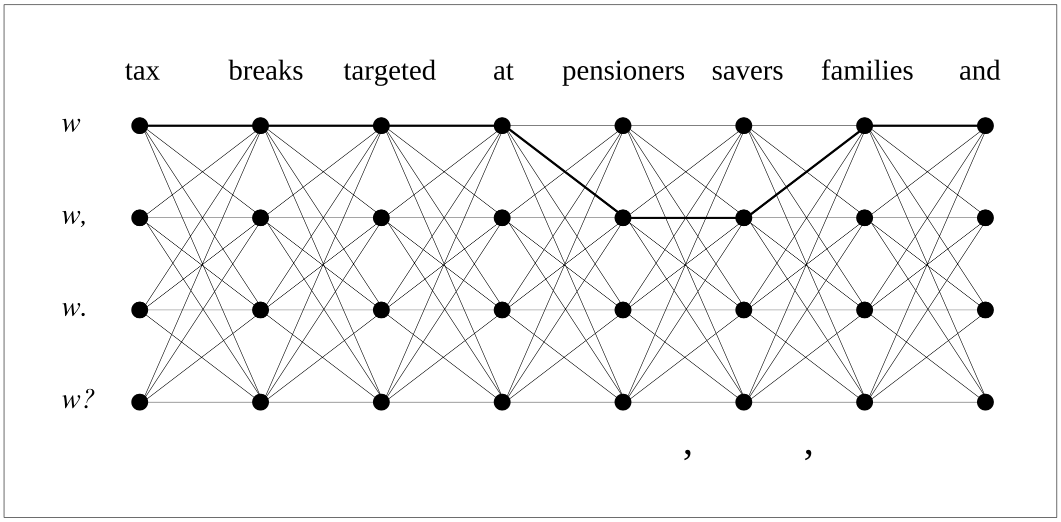

Figure 5.1 Viterbi search process for the generation of punctuation for an example reference transcription. The bold line depicts the best hypothesis. Punctuation marks at the bottom are generated according to this best hypothesis.

Figure 5.2 illustrates the overall procedure of punctuation generation for the reference transcription. The raw speech signal is time-aligned with the corresponding reference transcription. During this alignment process, the start time and the end time of each word are produced. Prosodic features are generated at the end of each word, and the probabilities $P(Y|R)$ are obtained from the prosodic feature model. $P(Y|W)$, the probability of the sequence of words and the possible punctuation marks are calculated from a statistical language model. The best hypothesis with punctuation marks is generated using the Viterbi search algorithm.

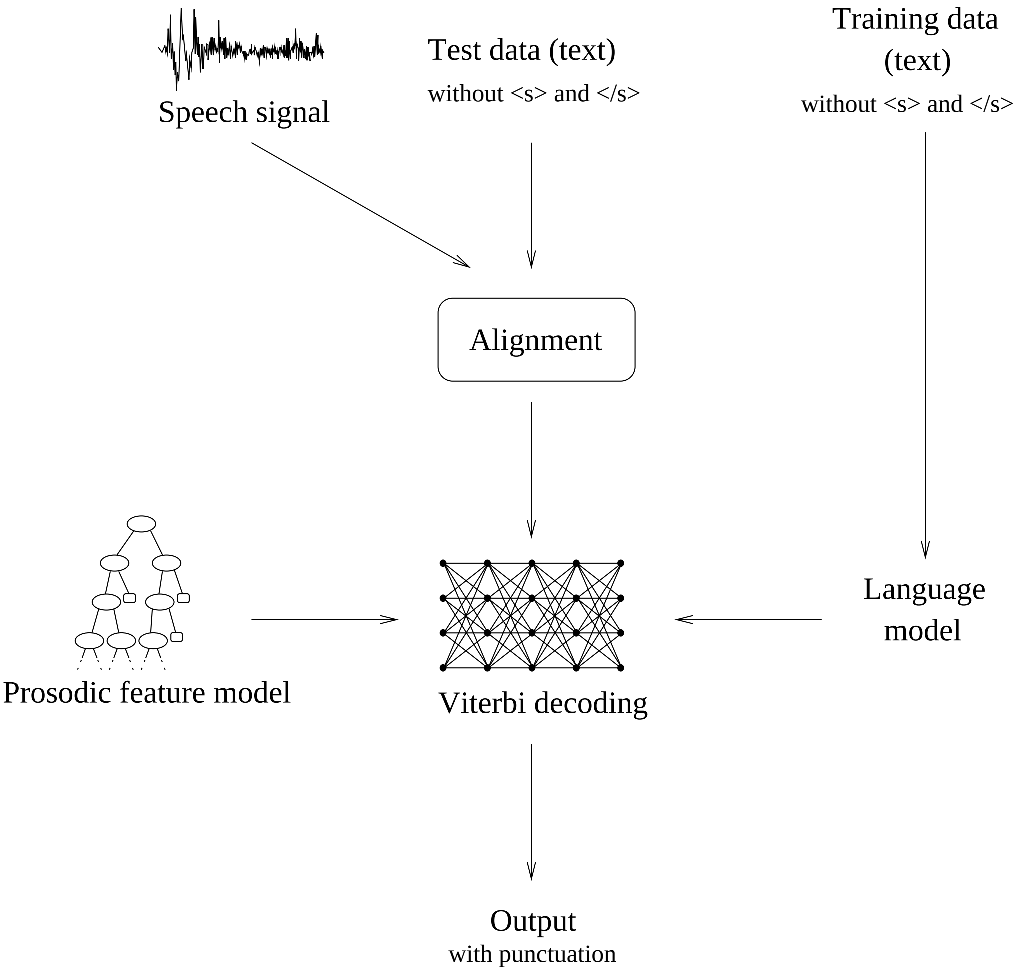

Figure 5.2 Overall procedures of punctuation generation for reference transcriptions

5.1.2 Automatic punctuation generation combined with speech recognition

The correlation between punctuation and pauses for read speech was investigated in [29]. These experiments showed that pauses closely correspond to punctuation marks. The correlation between pause lengths and sentence boundary marks was studied for broadcast news data in [47]. In their study, it was observed that the longer the pause duration, the greater the chance of a sentence boundary existing. Although some instances of punctuation do not occur at pauses, it is convenient to assume that the acoustic pronunciation of punctuation is silence. In this thesis, the pronunciation of punctuation marks is registered as silence in the pronunciation dictionary. The effectiveness of this assumption will be examined in Section 5.3.1.

A prosodic feature model to predict punctuation can be built by a classification tree. Probabilities from the prosodic feature model can then be incorporated by the re-scoring of multiple hypotheses each of which includes putative punctuation marks. The probability combination process can proceed as shown in Section 5.1.1.

Figure 5.3 illustrates the overall procedure in the generation of punctuation, when combined with speech recognition. Using language models and acoustic models, N-best hypotheses of speech recognition are produced from the raw speech signal. These N-best hypotheses contain punctuation marks. As these hypotheses contain the start time and the end time of every word contained in them, prosodic features are generated at the end of each word. Then the probability of prosodic features are measured from the prosodic feature model. The N-best hypotheses are re-scored using this probability of prosodic features, and the best hypothesis which includes punctuation marks is generated.

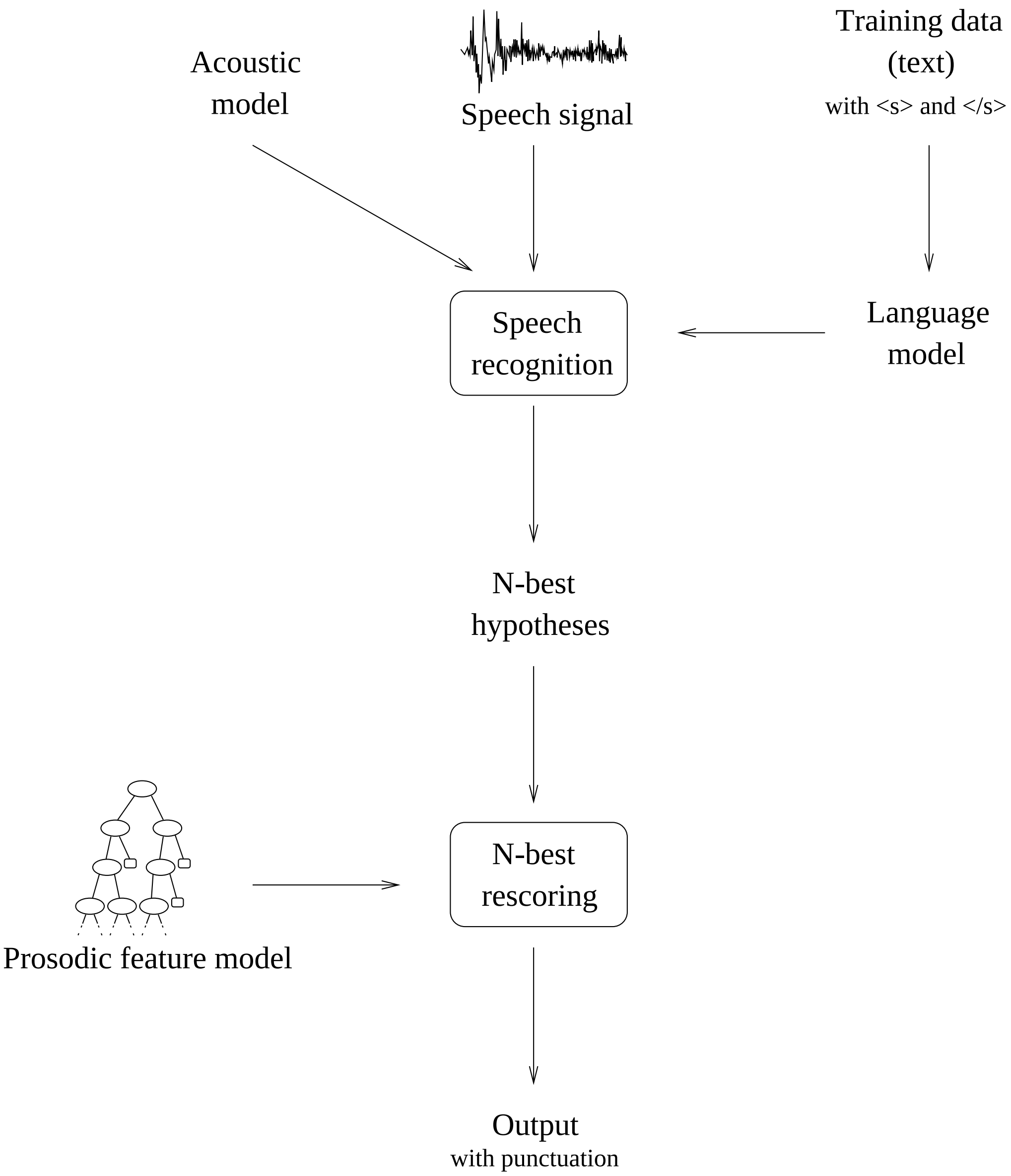

Figure 5.3 Overall procedures of punctuation generation combined with speech recognition

5.2 Experiments

As mentioned in Chapter 3, among many kinds of punctuation marks, this study is restricted to the examination of full stops, commas, and questions marks, because there are sufficient occurrences of these punctuation marks in the training data to be able to generate models and in the test data to measure the results accurately.

First, 4-gram LMs are produced by interpolating LMs trained on BNtext92_97 and DB98, using a perplexity minimisation method. The test data, TDB98, is provided as two separate parts. When automatic punctuation generation is performed for one part of the test data, the other part of the test data is used as the development set to estimate the LM mixture ratios. The LM mixture ratios are estimated using the 'interpolate' command in the CMU-Cam Toolkit. Details of the CMU-Cam Toolkit are given in [35]. Table 5.1 shows LM mixture ratios for each set of development data. When the whole of the test data was used for development data, the mixture ratios were estimated to be 0.3219 and 0.6781 for DB98 and BNtext92_97 respectively.

	LM mixture ratio	
Dev. Data	DB98	BNtext92_97
TDB98_1	0.3072	0.6928
TDB98_2	0.3460	0.6540

Table 5.1 LM mixture ratios determined by perplexity minimisation for each set of development data

5.2.1 Classification tree setup

Many easily computable prosodic features were investigated for Dialog Act (DA) classification in [80], for information extraction in [50], and for automatic topic segmentation in [82].

The prosodic features that were found to be most useful for these areas were applied in this thesis. By considering the automatic punctuation generation task and the contribution of each prosodic feature for DA classification, a set of 10 prosodic features were investigated for punctuation generation. Table 5.2 lists these 10 features. The first feature (Pau_Len) is a pause feature. The next feature (Dur_fr_Pau) is related to duration. Five other features (Avg_F0_L, Avg_F0_R, Avg_F0_Ratio, Cnt_F0_L, and Cnt_F0_R) are F0 related features, and the other three features (Eng_L, Eng_R, and Eng_Ratio) are energy features.

The end of each word is a possible candidate for punctuation, and so all prosodic features are measured at the end of a word. The window length is set at 0.2 seconds. The left window is the window to the left of the word end, and the right window, that to the right. "Good" F0 values are those greater than the minimum F0 (50Hz) and less than the maximum F0 (400Hz).

Name	Description
Pau_Len	Pause length at the end of a word
Dur_fr_Pau	Duration from the previous pause
Avg_F0_L	Mean of good F0s in left window
Avg_F0_R	Mean of good F0s in right window
Avg_F0_Ratio	Avg_F0_R/Avg_F0_L
Cnt_F0_L	No. of good F0s in left window
Cnt_F0_R	No. of good F0s in right window
Eng_L	RMS energy in left window
Eng_R	RMS energy in right window
Eng_Ratio	Eng_R/Eng_L

Table 5.2 Description of the prosodic feature set (Window length = 0.2 sec, 50Hz $\leq$ good F0 $\leq$ 400Hz)

A prosodic feature model is constructed using the Classification And Regression Tree (CART) [25] method. Prosodic features for the classification tree generation are measured from DB98 because it is the only database in the training set with acoustic data.

The CART of the prosodic feature model is constructed based on binary recursive splitting. The process is binary since parent nodes are split into two child nodes. In addition, this process is recursive since it can be repeated by treating each child node as a parent node. In order to define a specific application of CART, the following must be specified:

1. Generation of candidate queries and splitting criteria
2. Decision whether the recursive process is repeated or not
3. Assignment of a class to each terminal node

To split a node into two child nodes, candidate queries such as “Is pause length at the end of a word less than 0.0150 seconds?” are generated and the best candidate query is selected according to a splitting criteria. In this thesis, the entropy reduction criteria is used for the selection of the best candidate query. In the generation of candidate queries, the combination of features is not allowed in order to reduce the search space, and this makes the interpretation of queries easier.

Once the best splitting rule is found, the parent node is split and then the same procedure repeated for each child node. This process continues recursively until no further splitting is possible. The split is impossible when only one case remains in a particular node or when all the cases in that node are exactly the same.

Once a terminal node is generated, it must be assigned a label. One simple rule is used: the class with the greatest number of occurrences is given as the assignment of the class at a terminal node.

CART continues splitting until it classifies its training data with 100% accuracy. This CART fits the training data very well, but it does not guarantee the best performance for the test data because the CART is over-grown for the training data. The performance of a CART can be improved by pruning using the cross validation method.

In this thesis, the training data is divided into 10 roughly equal size parts. CART takes the first 9 parts of the data, constructs the largest possible tree, and uses the remaining 1/10 of the data to obtain the pruning variable. This process is called α-cut. The same process is then repeated on another 9/10 of the data while using a different 1/10 part for the pruning. The process continues until each part of the data has been used for the decision of the pruning variable.

After measuring 10 different pruning variables, a CART is generated for the whole of the training data. This CART is then pruned by the geometric mean of the 10 different pruning variables. Details of the CART generation method are given in [25].

Figure 5.4 depicts up to level-6 of the decision tree generated for the classification of punctuation marks as No-Punctuation (NP), comma (,), full stop (.) or question mark (?). The generated tree consists of 6161 nodes (3080 non-terminal nodes and 3081 terminal nodes). An internal node is depicted as an ellipse, and a terminal node is depicted as a rectangle. Each internal node explains its best splitting query which reduces the entropy most. If the condition of the query is met by input prosodic features, this input is moved to the left child node. If the condition is not met, the input is switched to the right child node.

The probability of the prosodic feature model for the input prosodic features is measured at a terminal node where the input features stop splitting. By pruning, there are some prosodic features allocated at this terminal node for each punctuation type. Based on the proportion of the occurrences of prosodic features for punctuation type to the total number of occurrences, the probability of prosodic feature model is calculated.

The overall contribution of different features can be measured by ‘feature usage’, which is the proportion of the number of times a feature is queried by the test data and can be measured by ‘feature appearance’, which is the number of times a feature is used as a classifying feature in non-terminal nodes. Table 5.3 shows the degree of overall contribution of each feature.

Figure 5.4 The generated decision tree for the classification of punctuation marks between No-Punctuation (NP), comma (,), full stop (.) and question mark (?)

Name	Feature appearance	Feature usage
Pau_Len	672	0.5799
Dur_fr_Pau	539	0.0230
Avg_F0_L	342	0.0246
Avg_F0_R	230	0.0363
Avg_F0_Ratio	261	0.0461
Cnt_F0_L	204	0.0429
Cnt_F0_R	230	0.0176
Eng_L	203	0.0038
Eng_R	160	0.0252
Eng_Ratio	239	0.2006

Table 5.3 Contribution of each feature for the CART trained by DB98 and tested by TDB98 (Feature usage: proportion of the number of times a feature is queried. Feature appearance: the number of times a feature is used as a classifying feature)

According to the measure 'feature usage', Pau_Len and Eng_Ratio are queried by about 78% of total queries. This measure accounts for the position of the feature in the tree. The higher the feature is used in the tree, the greater the feature usage is. In the classification tree depicted in Figure 5.4, the top node queries about Pau_Len, and the internal node at level-2 asks a query regarding Eng_Ratio.

Some classification statistics of the test data are shown in Table 5.4 in terms of the number of terminal nodes classified to each punctuation mark (#terminal) and the relative number of classifications for each punctuation mark in the training data (relative#).

Punctuation mark	#terminal	relative#
NP	788	0.9114
,	844	0.0347
.	1192	0.0530
?	257	0.0008

Table 5.4 Classification statistics of the test data (#terminal: number of terminal nodes classified to each punctuation mark; relative#: relative number of classification to each punctuation mark in the training data)

5.2.2 Results: Automatic punctuation generation for reference transcriptions

In order to generate punctuation marks for the reference transcription, three different systems were developed: a language model only system (S_LM), a prosodic model only system (S_CART), and the combination of these two systems (S_LM+CART). S_LM was trained on 185M words of transcriptions (BNtext92_97 and DB98). As these transcriptions contain punctuation marks, the language models trained on these transcriptions can predict the locations and types of punctuation marks based on word sequences which do not contain punctuation marks. 4-gram LMs are trained for S_LM. S_CART is generated on the 10 prosodic features described in Table 5.2 from a 100 hour broadcast news (DB98). More details about the database were given in Chapter 3.

The combination methodology of a prosodic feature model and a language model was explained in Section 5.1. Using the scale factor (α) which is the weighting given to the prosodic feature model, the relative importance of the prosodic feature model can be controlled. The scale factor is incorporated into the combination of these two systems i.e.

$$\alpha \times \log P(R|Y) + \log P(Y|W) \tag{5.11}$$

Table 5.5 summarises these three systems. In this section, the performances of these three systems are compared for punctuation generation for reference transcriptions.

System	Description
S_LM	Language model only
S_CART	Prosodic feature model only (by classification tree)
S_LM+CART	Combination of S_LM and S_CART

Table 5.5 Description of automatic punctuation generation systems for reference transcripts

The language model only system (S_LM) gives an F-measure of 0.5717 and an SER of 72.25%. When S_LM generates punctuation for the reference transcription, its precision (0.5966) is a little higher than its recall (0.5488). Surprisingly, the prosodic feature model alone (S_CART) outperforms S_LM by 0.0521 in F-measure and by 0.54% in SER. For S_CART, recall (0.7414) is much higher than precision (0.5383). These results show that S_CART produces a relatively high number of punctuation marks, but many of the generated punctuation marks need refinement.

As recall is much higher than precision for S_CART and precision is slightly higher than recall for S_LM, the two information sources, one from lexical information and the other from prosodic feature information, are expected to be complementary. By combining these two models, the results are greatly improved. The combined system (S_LM+CART) produces an F-measure of 0.7830 with an SER of 32.30%, a precision of 0.7638 and a recall of 0.8031. These results are obtained when the scale factor (α) of 2.0 is applied. The F-measure attains a maximum at a scale factor of 2.0. The SER attains a minimum at a scale factor of 1.8. The results of automatic punctuation generation for the reference transcript are summarised in Table 5.6.

System	Precision	Recall	F-measure	SER(%)
S_LM	0.5966	0.5488	0.5717	72.25
S_CART	0.5383	0.7417	0.6238	71.71
S_LM+CART (α=2.0)	0.7638	0.8031	0.7830	32.30

Table 5.6 Automatic punctuation generation results for reference transcripts (α = scale factor to the prosodic feature model; SER: Slot Error Rate)

The performance of S_LM+CART varies as the scale factor changes. Figure 5.5 describes how F-measure, precision, recall and SER change with the scale factor. The greater the scale factor for the prosodic feature model, the greater the recall because recall is much higher than precision for S_CART. Precision has a maximum value at a scale factor of 1.8. The F-measure attains a maximum of 0.7830 at a scale factor of 2.0. The SER attains a minimum of 32.12% at a scale factor of 1.8.

If the concept of scale factor is not introduced for this experiment, the probabilities from the language model and those from the prosodic feature model are combined by 1:1. When a scale factor of 1.0 is applied, the F-measure is 0.7668 and the SER is 34.16%. By the introduction of a scale factor, the F-measure is improved by 0.0162 (2.11% relative) and the SER by 2.04% (5.97% relative). Table 5.7 shows the results in detail. An example of punctuation generation output produced by S_LM+CART is shown in Figure 3 in the Appendix.

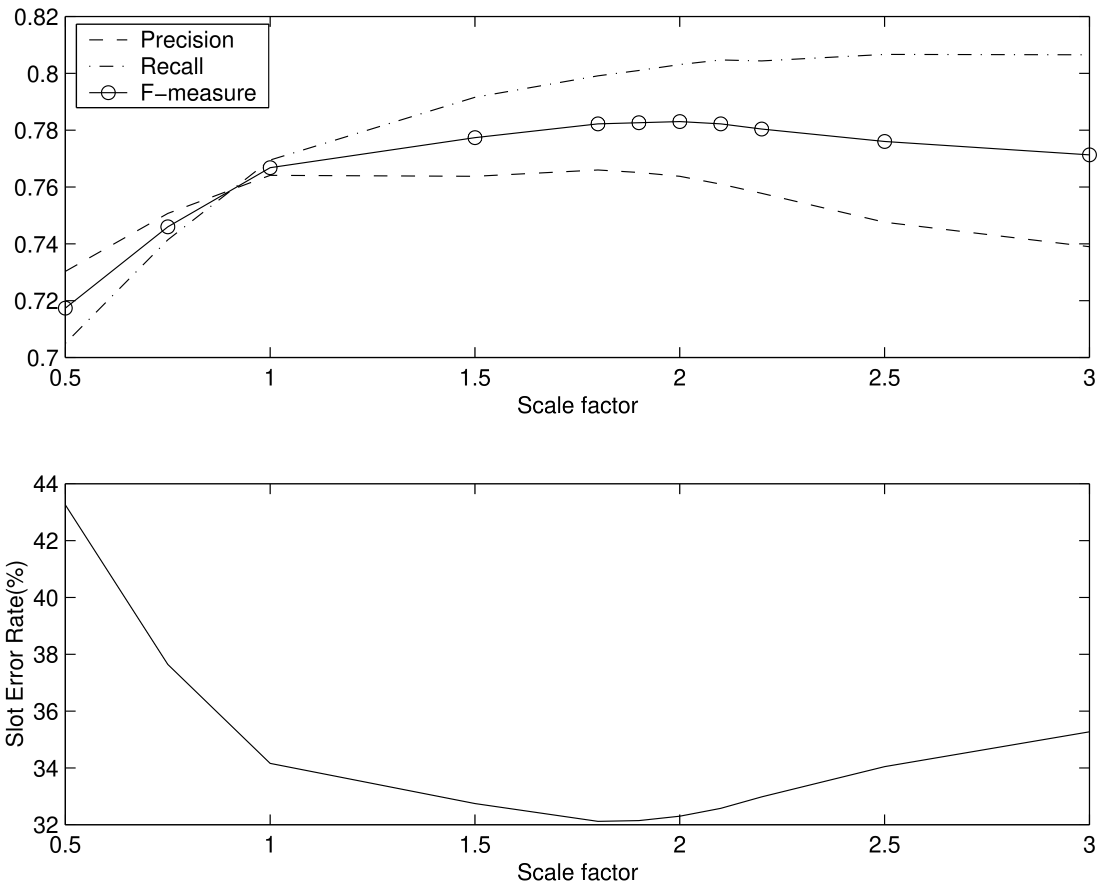

Figure 5.5 Automatic punctuation generation results of S_LM+CART with different scale factors

α	Precision	Recall	F-measure	SER(%)
0.5	0.7303	0.7050	0.7174	43.26
0.75	0.7507	0.7414	0.7460	37.64
1.0	0.7641	0.7695	0.7668	34.16
1.5	0.7638	0.7916	0.7774	32.75
1.8	0.7660	0.7991	0.7822	32.12
1.9	0.7651	0.8010	0.7826	32.15
2.0	0.7638	0.8031	0.7830	32.30
2.1	0.7610	0.8047	0.7822	32.58
2.2	0.7578	0.8044	0.7804	32.98
2.5	0.7476	0.8067	0.7760	34.05
3.0	0.7390	0.8065	0.7713	35.27

Table 5.7 Automatic punctuation generation results of S_LM+CART with different scale factors (α: scale factor to the prosodic feature model; SER: Slot Error Rate)

5.2.3 Results: Automatic punctuation generation combined with speech recognition

The HTK system [93] for Broadcast News (BN) transcription running under 10 times real time (10xRT) [69] was used for the task of combining automatic punctuation generation with speech recognition. The HTK 10xRT broadcast news transcription system is based on the HTK HMM toolkit. The first step of the system is a segmentation stage which converts the continuous input stream into segments with the aim of each segment containing data from a single speaker and a single audio type. Each segment is labelled as being either a wide-band or narrow-bandwidth signal.

The actual recogniser runs in two passes which both use cross-word triphone decision-tree state clustered HMMs with Gaussian mixture output distributions and a N-gram language model. The first pass uses gender-independent (but bandwidth-specific) HMMs with a 60k trigram language model to get an initial transcription for each segment. This transcription is used to determine the gender label for the speaker in each segment by alignment with gender-dependent HMMs. Sets of segments with the same gender/bandwidth labels are clustered for unsupervised Maximum Likelihood Linear Regression (MLLR) [59] adaptation. The MLLR transforms for each set of clustered segments are computed using the initial transcriptions of the segments and the gender-dependent HMMs used for the second pass. The adapted HMMs along with a 4-gram language model is used in the second stage of decoding and produces the final output.

Implementation details of the HTK BN transcription system (with few constraints on computing power) were given in [88, 89], and those of the HTK 10xRT BN transcription system were described in [69]. In order to speed up the full system, the 10xRT system uses simpler acoustic models and a simplified decoding strategy.

Using the HTK 10xRT system, speech recognition is performed first for TDB98. As punctuation is not considered at this stage, the test condition is the same as for the NIST 1998 Hub-4 broadcast news benchmark tests. The Word Error Rate (WER) of the speech recogniser is measured as 16.7%.

The HTK 10xRT BN transcription system reported 16.1% of overall WER for the NIST 1998 Hub-4 BN benchmark test [70]. The difference between the reported performance in [70] and the performance measured in this thesis is 0.6%. The system used in this thesis differs from the HTK 10xRT system used in the 1998 Hub-4 BN benchmark test in four aspects: the absence of a category-based language model [68], the amount of language model training data, the difference in vocabulary size, and the absence of the procedure to obtain more precise word start and end time information. This is explained further.

The HTK 10xRT system used in the 1998 Hub-4 BN benchmark test used a language model interpolated between a word 4-gram language model and a category based language model. However, the HTK 10xRT system used in this thesis does not use this category-based language model.

Another difference is the amount of training data for the construction of language models. According to the description of the HTK system in [69, 89], the size of the training text is about 260 million words. This training text covers BNtext92_97, DB98 and additional texts. There are also a difference in vocabulary size. The HTK system in [69, 89] used a 60K word size vocabulary, but the size of the vocabulary in the HTK 10xRT system used in this thesis is 108K.

In order to obtain more precise word start time and word end time information, the HTK 10xRT system used in the 1998 Hub-4 BN benchmark test removes silence models at the end of words. This improves the WER because it enhances the accuracy of the alignment process between a reference and a hypothesis. However, the removal of silence models at the end of words is not introduced in this thesis because the acoustic pronunciations of punctuation marks are registered as silence.

Table 5.8 shows speech recognition results under 3 different conditions. When punctuation is not included in training and test data, the WER of the speech recogniser (S_woP) is 16.71%. After including punctuation marks, the WER of the speech recogniser (S_Base) is increased to 22.73%. This degradation is caused by two factors: additional errors from other words due to the introduction of punctuation marks into the vocabulary, and errors in mis-recognising the punctuation marks themselves. In S_rmP, punctuation marks are generated by S_Base and these marks are then removed from the reference and the hypothesis. Using the degradation from S_woP to S_rmP, the error from other words due to adding punctuation marks to the vocabulary can be measured at 0.33%; the other factor is therefore measured at 5.69%.

System	WER	Remarks
S_woP	16.71	Punctuation excluded
S_Base	22.73	Punctuation included
S_rmP	17.04	Punctuation marks removed from reference and S_Base's result

Table 5.8 Speech recognition results (WER = Word Error Rate (%))

S_Base is used as the baseline automatic punctuation generation system with speech recognition. Using S_Base, 100 hypotheses are generated and re-scored on a segment basis using the prosodic feature model. After re-scoring, the best hypotheses for each segment are combined. Table 5.9 summarises these systems.

System	Description
S_Base	No re-scoring (baseline. WER = 22.73%)
S_H100	Final hypothesis from re-scored 100 hypotheses

Table 5.9 Description of automatic punctuation generation systems combined with speech recognition

The performances of S_H100 vary with the scale factor to prosodic model changes. Figure 5.6 describes how both the WER and the WER after punctuation is removed from reference and hypothesis (WER′) change according to scale factor. WER is minimised with a scale factor of 0.71, and WER′ is minimised with a scale factor of 0.79.

Although the amount of improvement in terms of WER is small, it is very important that these results show the possibility of performance enhancement in speech recognition using prosodic feature information. The prosodic feature model used in this thesis is focused only on the classification of punctuation marks. Therefore, the words apart from punctuation marks are categorised as a single group: No-Punctuation (NP). In spite of this simple categorisation for words which are not punctuation marks, the WER after punctuation is removed is also improved.

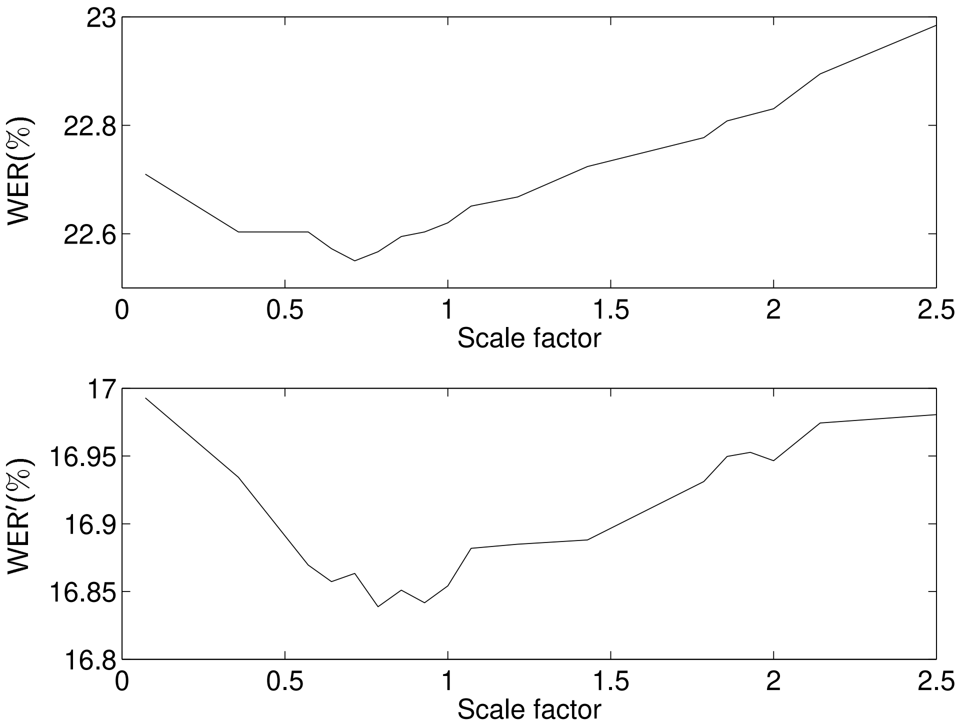

Figure 5.6 WER (Word Error Rate) and WER′ (WER after punctuation is removed from a reference and a hypothesis) of S_H100 with different scale factors

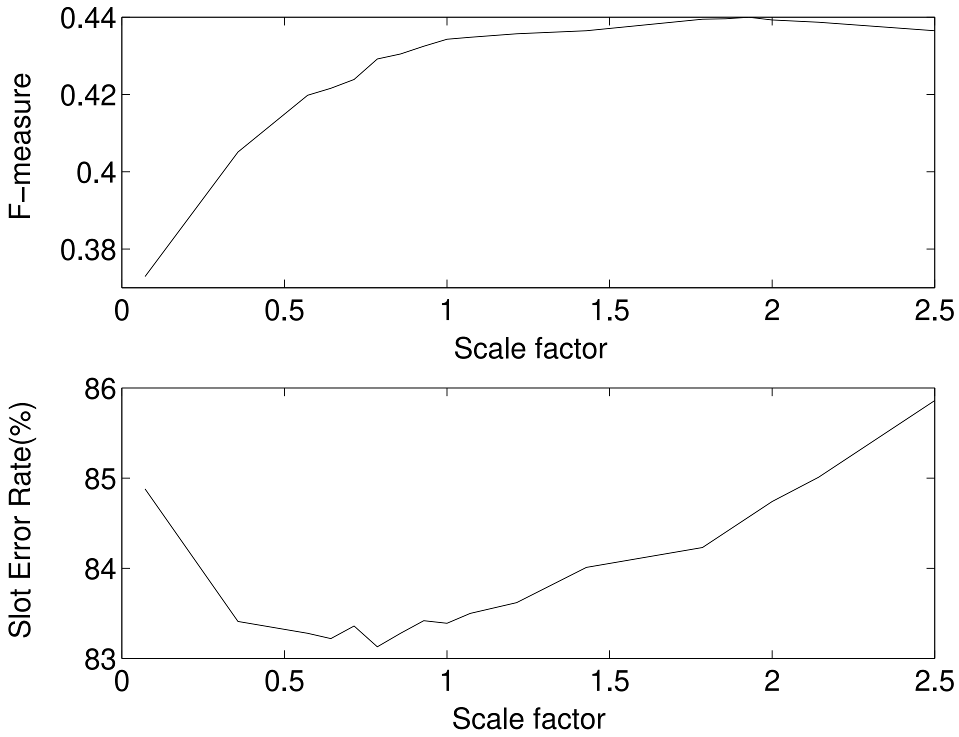

Figure 5.7 F-measure and SER of S_H100 with different scale factors

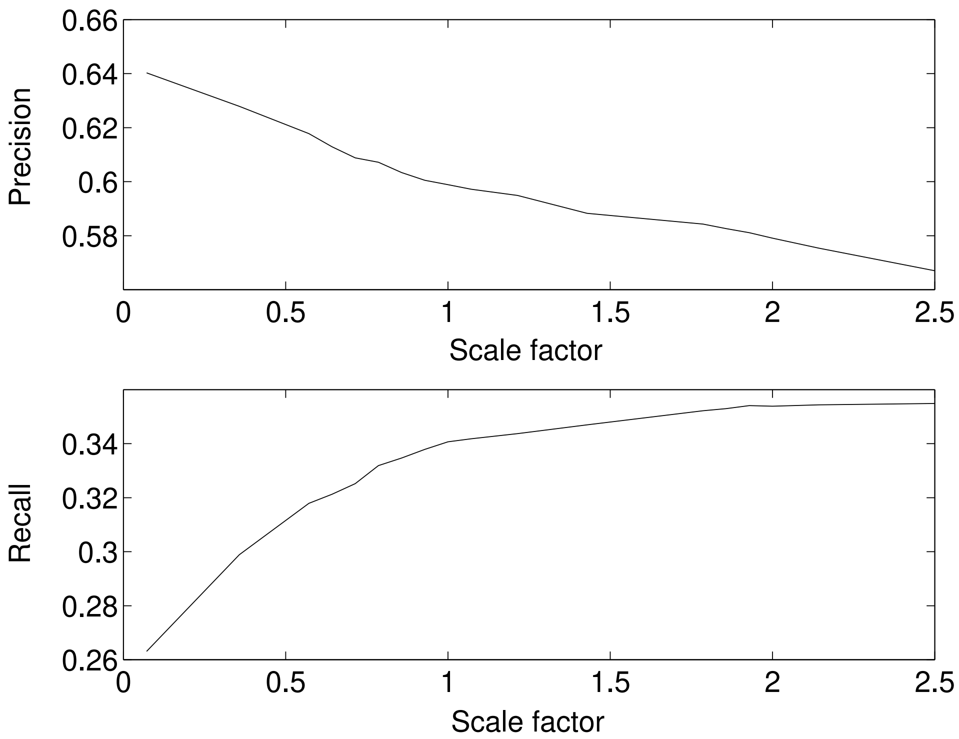

Figure 5.8 Precision and recall of S_H100 with different scale factors

Figure 5.7 shows the variation of F-measure and SER according to scale factor. Figure 5.8 shows that of precision and recall. The bigger the scale factor for the prosodic feature model, the bigger the recall and the smaller the precision is. The value of the F-measure attains its maximum of 0.4400 when the scale factor is 1.93. SER attains its minimum of 83.13% at the scale factor of 0.79.

If the re-scoring with prosodic feature model is not performed, the F-measure of the system is 0.3687, and the SER of the system is 85.02%. By the introduction of re-scoring with the prosodic feature model, the F-measure is improved by 0.0713 (19.34% relative) and the SER by 1.89% (2.22% relative).

Table 5.10 summarises these results. As the punctuation generation is combined with speech recognition, it is worth checking the result of punctuation generation when the best speech recognition performance is achieved. The precision, recall and F-measure are measured as 0.6072, 0.3319, and 0.4292 respectively at the scale factor of 0.79 when WER$'$ attains its minimum. At this scale factor, SER attains its minimum value of 83.13% too. These results show that the result of punctuation generation can be improved by the re-scoring of multiple hypotheses using a prosodic feature model while also improving speech recognition WER.

System	WER	WER$'$	Precision	Recall	F-measure	SER
S_Base	22.73	17.04	0.6425	0.2585	0.3687	85.02
S_H100 (α=0.79)	22.57	16.84	0.6072	0.3319	0.4292	83.13
S_H100 (α=1.93)	22.82	16.95	0.5811	0.3541	0.4400	84.57

Table 5.10 Results of automatic punctuation generation with speech recognition (WER: Word Error Rate (%); WER$'$: WER after removing punctuation from a reference and a hypothesis; SER: Slot Error Rate (%))

Table 5.11 shows the results of S_H100 with different scale factors. There are big differences between the values of precision and recall. The values of precision vary around 0.60 while the values of recall vary around 0.30. Comparing these results to the results of punctuation generation for the reference transcription shown in Section 5.2.2, the precision is satisfactory, but the recall is too low. This suggests that insufficient punctuation marks are generated in the hypotheses. As stated previously, in this thesis, the pronunciation of punctuation mark is assumed to be silence. This is only a rough approximation. This assumption will be analysed in Section 5.3.1.

α	WER	WER$'$	Precision	Recall	F-measure	SER
0.07	22.7098	16.9929	0.6403	0.2631	0.3729	84.88
0.36	22.6033	16.9343	0.6279	0.2989	0.4051	83.41
0.57	22.6033	16.8696	0.6178	0.3179	0.4198	83.28
0.64	22.5724	16.8573	0.6129	0.3213	0.4216	83.22
0.71	22.5500	16.8634	0.6088	0.3252	0.4239	83.36
0.79	22.5668	16.8388	0.6072	0.3319	0.4292	83.13
0.86	22.5948	16.8511	0.6034	0.3347	0.4305	83.28
0.93	22.6033	16.8418	0.6005	0.3379	0.4325	83.42
1.00	22.6201	16.8542	0.5989	0.3407	0.4343	83.39
1.07	22.6509	16.8819	0.5972	0.3418	0.4348	83.50
1.21	22.6678	16.8850	0.5949	0.3437	0.4357	83.62
1.43	22.7238	16.8881	0.5883	0.3470	0.4365	84.01
1.79	22.7771	16.9312	0.5843	0.3522	0.4395	84.23
1.86	22.8080	16.9497	0.5826	0.3530	0.4396	84.40
1.93	22.8192	16.9528	0.5811	0.3541	0.4400	84.57
2.00	22.8304	16.9466	0.5791	0.3539	0.4393	84.74
2.14	22.8949	16.9744	0.5754	0.3544	0.4387	85.01
2.50	22.9846	16.9806	0.5670	0.3549	0.4365	85.86

Table 5.11 Automatic punctuation generation results of S_H100 with different scale factors (α: scale factor; WER: Word Error Rate (%); WER$'$: WER after removing punctuation marks; SER: Slot Error Rate (%))

5.3 Error analysis

The pronunciation of punctuation marks was assumed to be silence. In addition, pause length was shown to be the most useful prosodic feature for punctuation mark generation using the prosodic feature model. In this section, the effectiveness of the assumption for the pronunciation of punctuation marks is examined, and the effectiveness of the prosodic feature model constructed by CART is measured.

The punctuation generation system with speech recognition reported relatively low recall compared to its precision. The results of the punctuation generation system with speech recognition are estimated and its actual results are compared with this estimation. In addition, a different punctuation generation system which does not use the assumption for the pronunciation of punctuation marks is proposed, and its results are compared with those of the punctuation generation system with speech recognition. Finally, the variation between annotators for punctuation marks is measured.

5.3.1 The effectiveness of the assumption for punctuation mark pronunciation

The pronunciation of punctuation marks was assumed to be silence. In this section, the effectiveness of this assumption is examined using TDB98.

The reference word sequence of TDB98 is time aligned with its acoustic data. This word sequence does not contain any punctuation mark. Then, the duration of the models 'sp' and 'sil' are measured at the end of each word. Table 5.12 shows the ratio of presence of silence for each punctuation mark type. About 90% of full stops and question marks are related to silence, but pauses do not exist at about 40% of commas. In addition, pauses are measured at the end of about 15% of words where no punctuation is located.

Punctuation mark	Ratio of presence of silence(%)
NP	15.42 (4352/28218)
,	60.58 (948/1565)
.	88.63 (1590/1794)
?	91.84 (45/49)

Table 5.12 Ratio of presence of silence for each punctuation mark type (NP: No-Punctuation)

The pause lengths have different distributions according to the type of punctuation mark. Figure 5.9 shows the relative frequency of pause length according to the type of punctuation mark. Each pause length is counted and added at 0.05 second intervals. The distribution of pause length is different for each punctuation mark. Normally, the pause lengths at commas are shorter than those at full stops and question marks.

5.3.2 The effectiveness of the prosodic feature model

Pause length was shown to be the most useful prosodic feature for punctuation mark generation using the prosodic feature model. In Table 5.6, the prosodic feature model-only punctuation generation system (S_CART) reported an F-measure of 0.6238 with a precision of 0.5383, a recall of 0.7417 and a SER of 71.71% for the reference transcription of TDB98. In this section, the effectiveness of the prosodic feature model constructed by CART is measured using the ratio of presence of silence illustrated in Table 5.12.

Assume that a punctuation mark is generated at every pause with the same type of punctuation mark as in the reference. From Table 5.12, the numbers of correct slots, deletion errors and insertion errors are counted as 2583 (948 + 1590 + 45), 825 ((1565-948) + (1794-1590) + (49-45)), and 4352 respectively, if it is assumed that there are no substitution errors. From these numbers, F-measure, recall, precision and SER are measured as 0.4995, 0.7579, 0.3725, and

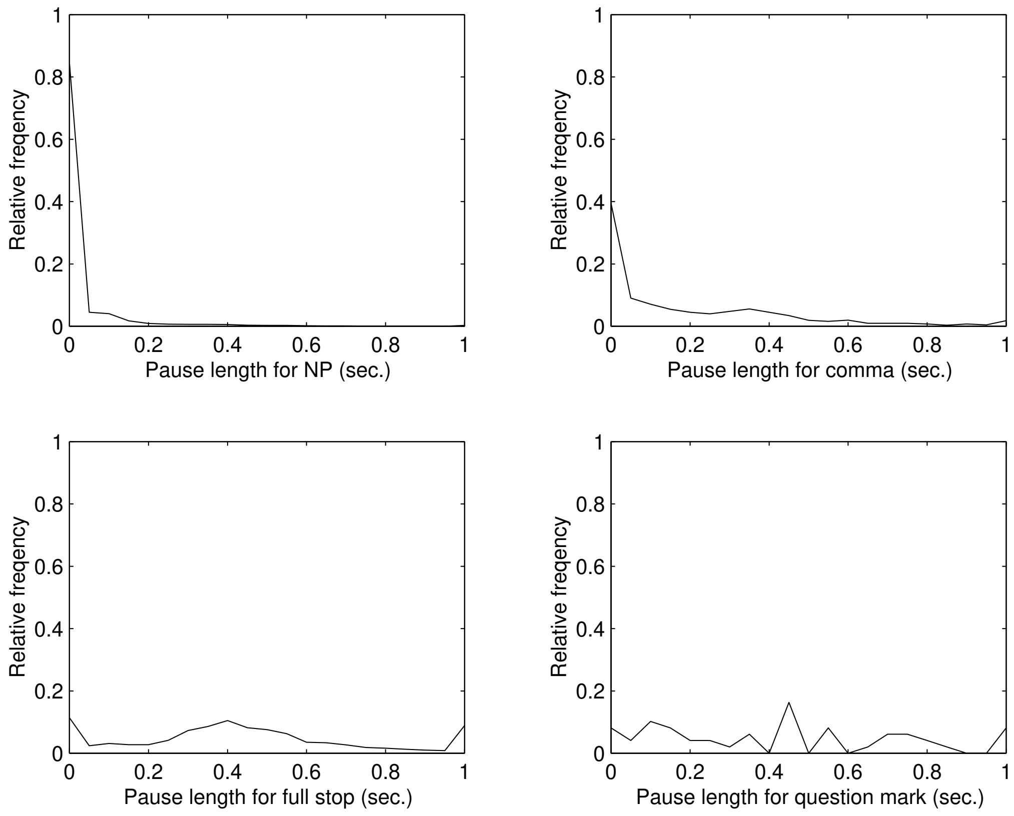

Figure 5.9 Distribution of pause length according to the type of punctuation mark (NP: No-Punctuation, Interval: 0.05 sec.)

151.91% respectively. Considering the differences in F-measure and SER, the prosodic model-only punctuation generation system (S_CART) produced good results using pause length and other prosodic features.

5.3.3 Estimation: Result of the punctuation generation system with speech recognition

The punctuation generation system with speech recognition reported relatively too low recall compared to its precision. In this section, the results of the punctuation generation system with speech recognition are estimated and its actual results are compared with this estimation. In order to remove the effects of prosodic features, the results of the punctuation generation system with speech recognition which does not use the re-scoring by the prosodic feature model (S_Base) are estimated from the results of the language model-only punctuation generation system for reference transcription (S_LM). S_LM reported a precision of 0.5966 and a recall of 0.5488 for reference transcripts with 1779 correct slots, 323 substitution errors and 879 insertion errors. S_Base reported a precision of 0.6425 and a recall of 0.2585 with 832 correct slots, 122 substitution errors and 341 insertion errors.

The end of each word is a possible candidate for a punctuation mark. Denote i-th word as W_i and a punctuation mark at the end of W_i as P_i (P_i can be No-Punctuation). For each P_i in the hypothesis of S_Base, there are 8 different cases depending on whether P_i is a punctuation mark in the reference and in the hypothesis of S_LM, and whether there is a speech recognition error in W_i and W_{i+1}. Table 5.13 summarises these 8 cases.

Condition	Case number							
	1	2	3	4	5	6	7	8
Is P_i a punctuation mark in reference?	Y	Y	Y	Y	N	N	N	N
Is P_i a punctuation mark in hypothesis of S_LM?	Y	Y	N	N	Y	Y	N	N
Is either of W_i or W_{i+1} a speech recognition error?	Y	N	Y	N	Y	N	Y	N

Table 5.13 Summary of 8 different cases for punctuation marks in the hypothesis of S_Base

The punctuation generation system with speech recognition uses 'silence' as the pronunciation of punctuation marks. Therefore, it is required that a pause should be placed at P_i to produce a punctuation mark at the position of P_i in the hypothesis produced by S_Base. Introduce an assumption that word sequences which contain speech recognition errors follow the overall statistics of TDB98. The number of punctuation marks produced in the hypothesis of S_Base can be estimated for each case as follows:

1. Case 1:

   (1.a) Number of cases in which there is a punctuation mark at P_i in reference and in hypothesis of S_LM: number of correct slots and substitution errors of S_LM = 2,102

   (1.b) Probability that there is a speech recognition error of S_Base at P_i or P_{i+1}: 1-(1-WER$'$ of S_Base)^2 = 0.3118

   (1.c) Probability of pause existence between W_i and W_{i+1}, at least one of which is speech recognition error by S_Base: total number of pause / total number of words = 0.2193

   (1.d) Probability of punctuation generation by the LM between W_i and W_{i+1}, at least one of which is speech recognition error by S_Base: total number of generated punctuation marks by S_LM / total number of words = 0.0943

   The total number of generated punctuation marks by S_Base for case 1: (1.a) $\times$ (1.b) $\times$ (1.c) $\times$ (1.d) = 14. These are the correct slots or substitution errors of S_Base.

2. Case 2:

(2.a) Number of cases in which there is a punctuation mark at P_i in reference and in hypothesis of S_LM: same as in (1.a) = 2,102

(2.b) Probability that there is no speech recognition error at W_i and W_{i+1}: (1-WER$'$ of S_Base)^2 = 0.6882

(2.c) Probability of pause existence at punctuation mark: total number of pauses at punctuation marks / total number of punctuation marks = 0.7579

The total number of generated punctuation marks by S_Base for case 2: (2.a) $\times$ (2.b) $\times$ (2.c) = 1,096. These are the correct slots or substitution errors of S_Base.

3. Case 3:

(3.a) Number of cases in which there is a punctuation mark at P_i in reference but not in hypothesis of S_LM: number of deletion errors of S_LM = 1,139

(3.b) Probability of punctuation generation between W_i and W_{i+1}, at least one of which is speech recognition error by S_Base: (1.b) $\times$ (1.c) $\times$ (1.d) = 0.0064

The total number of generated punctuation marks by S_Base for case 3: (3.a) $\times$ (3.b) = 7. These are the correct slots or substitution errors of S_Base.

4. Case 4:

No punctuation mark is generated by S_LM between W_i and W_{i+1}. Punctuation cannot be generated for the same word sequence by S_Base.

5. Case 5:

(5.a) Number of cases in which there is no punctuation mark at P_i in reference but there is in hypothesis of S_LM: number of insertion errors of S_LM = 879

(5.b) Probability of punctuation generation between W_i and W_{i+1}, at least one of which is speech recognition error by S_Base: same as in (3.b) = 0.0064

The total number of generated punctuation marks by S_Base for case 5: (5.a) $\times$ (5.b) = 6. These are the insertion errors of S_Base.

6. Case 6:

   (6.a) Number of cases in which there is no punctuation mark at P_i in reference but there is in hypothesis of S_LM: number of insertion errors of S_LM = 879

   (6.b) Probability that there is no speech recognition error at W_i and W_{i+1}: same as in (2.b) = 0.6882

   (6.c) Probability of pause existence at the position where no punctuation is: total number of pauses at the position where no punctuation is / total number of NP = 0.1543

   The total number of generated punctuation marks by S_Base for case 6: (6.a) × (6.b) × (6.c) = 93. These are the insertion errors of S_Base.

7. Case 7:

   (7.a) Number of cases in which there is no punctuation mark at P_i in reference or in hypothesis of S_LM: total number of words - number of hypothesised punctuation marks by S_LM - number of deletion errors of S_LM = 27,475

   (7.b) Probability of punctuation generation between W_i and W_{i+1}, at least one of which is speech recognition error by S_Base: same as in (3.b) = 0.0064

   The total number of generated punctuation marks by S_Base for case 7: (7.a) × (7.b) = 176. These are the insertion errors of S_Base.

8. Case 8:
   No punctuation mark is generated by S_LM between W_i and W_{i+1}. Punctuation cannot be generated for the same word sequence by S_Base.

Based on the estimation for each case, the total number of correct slots, substitution errors and insertion errors of S_Base are estimated to be 945, 172, and 275 respectively, if it is assumed that the ratio of correct slot to substitution errors is the same as S_LM. According to these numbers, recall and precision are estimated as 0.2916 and 0.6789 respectively. These estimations for the recall and the precision are only a little higher than their actual values, in spite of the rough estimations (by 0.033 for the recall and by 0.036 for the precision). The difference between the estimated and the actual values for correct slots, substitution errors and insertion errors are 113, 50 and 66 respectively, in spite of the rough estimation. From the estimation in this section, it is concluded that the recall of the punctuation generation system with speech recognition is reasonable, as long as it uses the assumption for the pronunciation of punctuation marks.

5.3.4 Comparison with the system which does not use the assumption for the pronunciation of punctuation marks

In this section, a different punctuation generation system which does not use the assumption for the pronunciation of punctuation marks is proposed, and its results are compared with those of the punctuation generation system with speech recognition.

The proposed system (S_1Best) generates punctuation marks from the 1-best output of a speech recogniser. In this speech recogniser, none of the punctuation marks is registered in its pronunciation dictionary. In addition, its language model is trained on a training text which does not contain any punctuation mark. As a result, this speech recogniser does not produce any punctuation mark. The 1-best output is time aligned. Based on the time alignment information, prosodic features are generated. As in the approach applied in the punctuation generation for reference transcripts in Section 5.1.1, the sequence of punctuation marks for this 1-best output is searched for using the prosodic feature model and an LM trained on texts which contain punctuation marks.

The trends of F-measure and SER of S_1Best are similar to the automatic punctuation generation system for reference transcription (S_LM_CART). The SER of S_1Best minimises at an alpha of 1.90 and its F-measure maximises at an alpha of 2.10. The results of S_1Best are measured at 2.10 and those of the punctuation generation system with speech recognition (S_H100) are measured at an alpha of 1.93 where its F-measure maximises. Table 5.14 summarises the descriptions of these systems.

System	Description	α
S_H100	Punctuation generation system with speech recognition	1.93
S_1Best	Punctuation generation system from 1-best output	2.10

Table 5.14 Summary of the punctuation generation systems used in performance comparison (α: scale factor to the prosodic feature model)

Table 5.15 compares the results of S_1Best with those of S_H100. As S_1Best uses the 1-best output of the speech recogniser without punctuation marks, WER$'$ of S_1Best is not affected by degradation due to the inclusion of punctuation marks into the vocabulary. S_1Best shows a better performance in terms of F-measure and WER$'$, but poorer in terms of WER and SER. If precision is more important than recall, S_H100 is the better system, but if recall is more important than precision, S_1Best is shown to be better.

System	WER	WER′	Precision	Recall	F-measure	SER
S_H100	22.82	16.95	0.5811	0.3541	0.4400	84.57
S_1Best	23.08	16.71	0.5329	0.4304	0.4762	88.32

Table 5.15 Comparison of results of S_1Best with S_H100 (WER: Word Error Rate (%); WER′: WER after punctuation is removed from a reference and a hypothesis; SER: Slot Error Rate (%))

As S_1Best does not assume that the pronunciation of punctuation marks is silence, S_1Best may produce punctuation marks at no-silence. The word sequence of the 1-best output was time aligned with its acoustic data. Then, the duration of the models 'sp' and 'sil' were measured at the end of each word. 58% of the hypothesised punctuation marks produced by S_1Best were found to be not related to silence. This rather high percentage is somewhat surprising. As a substantial number of these hypothesised punctuation marks are in error, it is assumed that the alignment process is affected by speech recognition errors.

5.3.5 The variations of punctuation marks between annotators

The use of punctuation is documented in manuals and in hand-books such as in [9, 79]. However, the style of punctuation varies between writers and between areas of texts [29]. In addition, punctuation marks are used to change the meaning of sentences. In this section, the variations of putting punctuation marks between annotators are measured.

The first 1000 words of TDB98 is prepared for this experiment. As capitalisation information gives cues to the location of sentence boundaries, these 1000 words are de-capitalised. Three English native speakers were asked to add punctuation marks between words wherever the punctuation marks are necessary. Only commas, full stops and question marks are permitted as punctuation marks. Although this experiment is performed with a small size text and a small number of annotators, it gives the general idea about the variations of punctuation marks between different annotators for the domain of broadcast news. Table 5.16 summarises these experimental conditions.

Condition	Description
Text source	First 1000 words in TDB98
Writing style	Single case. No punctuation mark
Annotator	Three native British English speakers

Table 5.16 Summary of the conditions of the experiment to measure the variations in putting punctuation marks between annotators.

In the provided reference transcription of TDB98, there are 43 commas and 54 full stops between the first 1000 words of TDB98. Table 5.17 shows the differences between the punctuation marks in the provided reference transcription and each annotator's transcription. These differences are measured in terms of precision, recall, F-measure and SER, regarding the provided transcription as the reference and each annotator's transcription as the hypothesis. On average, about 28% of punctuation marks conflict.

Source of hypothesis text	Precision	Recall	F-measure	SER(%)
Annotator 1	0.7558	0.6701	0.7104	49.48
Annotator 2	0.7158	0.7010	0.7083	47.42
Annotator 3	0.7448	0.7371	0.7409	44.85

Table 5.17 The difference of putting punctuation marks between the provided reference transcription and each annotator's transcription. The provided transcription is regarded as the reference and each annotator's transcription as the hypothesis. (SER: Slot Error Rate)

Table 5.18 shows the variations in punctuation between annotators. These variations are measure in terms of precision, recall, F-measure and SER, regarding an annotator's text as the reference and another annotator's text as the hypothesis. On average, about 29% of punctuation marks conflict.

Source of text		Results of variations			
Reference	Hypothesis	Precision	Recall	F-measure	SER(%)
Annotator 1	Annotator 2	0.6421	0.7093	0.6740	60.47
Annotator 1	Annotator 3	0.7188	0.8023	0.7582	44.19
Annotator 2	Annotator 3	0.6979	0.7053	0.7016	49.47

Table 5.18 Variations in punctuation between annotators. Results of variations are measured regarding an annotator's text as the reference and another annotator's text as the hypothesis. (SER: Slot Error Rate)

In this section, the variations of punctuation marks between annotators are measured. The amount of this variation is quite substantial. Even though the acoustic data for the text is provided when the reference text is transcribed, the punctuation marks in the provided reference text are not a perfect measure. This variation may partly account for reported punctuation generation errors.

5.4 Summary

In this chapter, an automatic punctuation method which generates punctuation marks simultaneously with speech recognition output has been presented. This system produces multiple hypotheses and uses prosodic features to re-score the hypotheses. Given the reference transcription, using prosodic information alone outperforms using lexical information alone. As these two information sources are shown to be complementary, further improvements can be achieved by combining these two information sources. When punctuation is generated simultaneously with speech recognition output, the F-measure can be improved up to 0.44 by utilising prosodic information. At the same time, small reductions in WER are achieved.

Chapter 6
Automatic capitalisation generation

In this chapter, another important area of transcription readability improvement, automatic capitalisation generation, is discussed. Two different systems are proposed for this task. The first is a slightly modified speech recogniser. In this system, every word in its vocabulary is duplicated: one in a capitalised form and the other in a de-capitalised form. In addition, its language model is re-trained on mixed case texts. The other system is based on NE recognition and punctuation generation since most capitalised words are the first words in sentences or NE words.

In order to compare the performance of the proposed systems, experiments of capitalisation generation are conducted when every procedure is fully automated. The system based on NE recognition and punctuation generation shows better results in WER, in F-measure and in SER. The contribution of each procedure in the system based on NE recognition and punctuation generation is examined, and the performance of this system is examined for the additional clues: reference word sequences, reference NE classes, and reference punctuation marks. Experimental results show that this system is robust to NE recognition errors and that the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation.

In Section 6.1, the two different automatic capitalisation generation systems are described. Experimental results are then shown in Section 6.2 and the results are analysed in Section 6.3. Finally, this chapter is concluded in Section 6.4.

6.1 Capitalisation generation

Standard transcriptions of speech lack most capitalisation and punctuation. As already mentioned in Table 2.3 for a 3 hour broadcast news transcription (TDB98), 15.26% of total words are capitalised words. The proper capitalisation of words would improve the readability of transcriptions substantially.

Many commercial implementations of automatic capitalisation generation are provided with word processors. In these implementations, grammar and spelling checkers of word processors generate suggestions about capitalisation. A typical example is one of the most popular word processors, Microsoft Word.

An experiment of automatic capitalisation generation was conducted using Microsoft Word 2000 for the first 10.7% words of TDB98 (3882 words, 468 of which are capitalised). As it provides suggestions about both grammar and spelling, its suggestions are checked manually and only suggestions regarding capitalisations are accepted. Table 6.1 shows the results of this experiment.

System	Precision	Recall	F-measure	SER(%)
MS Word 2000	0.9987	0.8045	0.8911	19.66

Table 6.1 Results of capitalisation generation using Microsoft Word 2000 for a part of TDB98 (SER: Slot Error Rate)

The implementation of the capitalisation generation in Microsoft Word was described in Section 2.2. According to the description in [77] and its capitalisation generation output for the part of TDB98, capitalisation of words which are not first words in sentences seems to be processed by dictionary look-up. When a word is entered in all lower case, the capitalisation is applied for the word to have the greatest consistency in matching the capitalisation.

With this dictionary look-up method, ambiguous words such as 'bill' cannot be dis-ambiguated. As seen in Section 1.2, in a sentence like "President bill Clinton says", 'bill' should be capitalised: the error occurs because the word 'bill' is more frequently used as a statement of account in a de-capitalised form rather than a person's name. Dis-ambiguation of the capitalisation type of words which can have more than one type can be achieved by using context information.

In this chapter, two different automatic capitalisation generation systems are presented. The first system is a slightly modified speech recogniser. In this system, every word in its vocabulary is duplicated: one in a capitalised form and the other in a de-capitalised form. In addition, its language model is re-trained on mixed case texts. This system will be presented in Section 6.1.1. The other system is based on NE recognition and punctuation generation, since most capitalised words are first words in sentences or NE words. This system will be presented in Section 6.1.2.

These systems examine the three types of capitalisation: all characters of a word are capitalised (All_Cap), only first character of a word is capitalised (Fst_Cap), and every character of a word is de-capitalised (No_Cap). The categories of capitalisation types have already been described in Table 1.1. Details of data preparation regarding capitalisation were given in Section 3.1.4. The performance of these two systems with every procedure being fully automated, will be compared in Section 6.2.

6.1.1 Automatic capitalisation generation by modifications of speech recogniser

The method of automatic capitalisation generation presented in this section is a slightly modified form of a conventional speech recogniser. As the aim of speech recognition is to find out only the best word sequences for the given speech signal, speech recognition systems do not normally recognise capitalisation of words. Therefore, the words registered in a vocabulary and a pronunciation dictionary are not case-sensitive in a conventional speech recognition system. In addition, it is not necessary to train language models of this system on case sensitive texts.

Slight modifications to a conventional speech recognition system, however, can produce case sensitive outputs. The following three modifications are required:

1. Every word in its vocabulary is duplicated three times for the three different capitalisation types (All_Cap, Fst_Cap, and No_Cap).
2. Every word in its pronunciation dictionary is duplicated with its pronunciation in the same way as used for the vocabulary duplication.
3. The language model is re-trained on mixed case texts.

This method is a good way to obtain capitalisation automatically. However, it faces the following two problems:

1. Distortion of LM

   In many cases, first words in sentences are non-NEs. Most of these words are not capitalised if they are used in the middle of a sentence. Therefore, a substantial number of word sequences counted at sentence boundaries are erroneous because a capitalised word and a de-capitalised word are regarded as different words even if they have the same character sequence.

2. Sparser LM

   Due to the limited amount of training data, many of the possible word sequences in test data are not observed in training data. As the size of vocabulary is increased by the duplication, LMs are sparser and estimating probabilities of word sequences becomes more difficult. In addition, the searching space is widened because of the increased size vocabulary.

These two problems will be analysed quantitatively in Section 6.2.1.

Figure 6.1 illustrates the overall procedures of the capitalisation generation system, modified from a conventional speech recognition system. Every word in the pronunciation dictionary of a conventional speech recogniser is duplicated. As an LM is trained on case sensitive training data, this LM is sparser than that used by the conventional speech recogniser. The same acoustic score is measured for duplicated words, since they have the same pronunciations. However, hypotheses can be generated using the different LM scores. Speech recognition is performed, and the best hypothesis which includes capitalisation is generated.

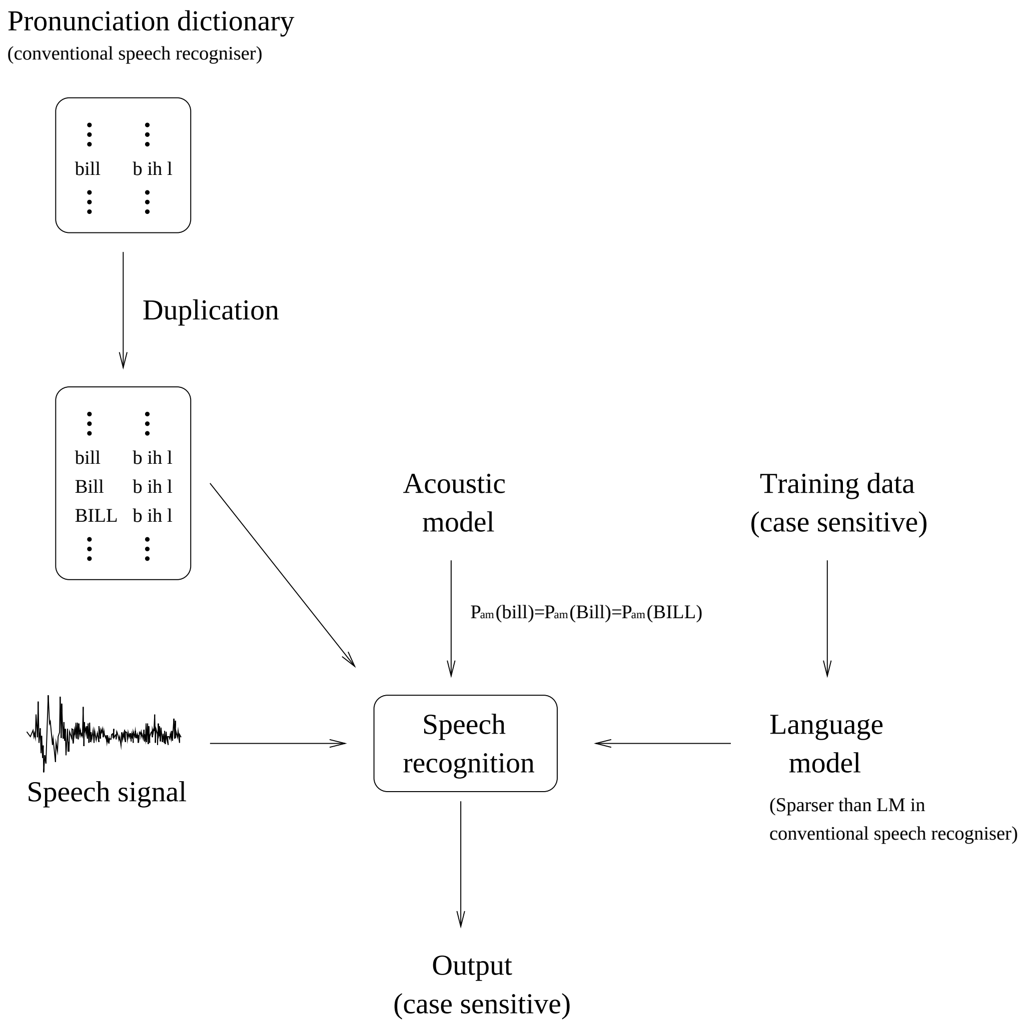

Figure 6.1 Overall procedures of the capitalisation generation system modified from speech recogniser

6.1.2 Automatic capitalisation generation based on NE recognition and punctuation generation

In TDB98, 15.26% of total words are capitalised. Most capitalised words are first words in sentences or NE words. As the average number of words in a sentence is 16.87, 5.23% of total words are first words in sentences. 80.45% of NE words are capitalised. Among non-NE words which are not first words in sentences, 2.32% of words are capitalised. For statistics of capitalisation for TDB98, see Table 2.3.

The fact that most capitalised words are first words in sentences or NE words motivates a capitalisation generation method based on NEs and sentence boundaries. The method of capitalisation generation presented in this section is based on NE recognition and punctuation generation. The simplest way to achieve capitalisation generation is to capitalise the first characters of words which are first words in sentences and the first characters of NE words whose NE classes are 'ORGANIZATION', 'PERSON', or 'LOCATION', followed by capitalisation of initials.

The results of capitalisation generation are improved by using a frequency table counted from training texts. Some NE words are used in de-capitalised forms and some non-NE words are used in capitalised forms. Also, all characters should be capitalised in some first words in sentences. Many of these capitalisation types are corrected by looking-up in a frequency table of words based on NE classes.

Further improvement is achieved by using context information to dis-ambiguate the capitalisation types of words which have more than one capitalisation type such as the word 'bill'. The context information about capitalisation generation is encoded in a set of simple rules rather than the large tables of statistics used in stochastic methods. The ideas used in the development of the rule-based NE recognition system are applied in the automatic generation of these rules for capitalisation generation.

Six rule templates are used for the generation of bigram rules for capitalisation generation. These six rule templates are shown in Table 6.2. As with the rule templates in NE recognition, rule templates consist of pairs of characters and a subscript. w, t, c denote that templates are related to words, NE classes and capitalisation types, respectively. Subscripts show the relative distance from the current word; that is 0 means the current word, -1 means the previous word and 1 means the next word. Each rule template has its own applicable range where the conditions of the rule are met. For these six rule templates, the range of rule application is set to be the current word only. Rule templates have one more slot at the end. This indicates the number of the capitalisation type of the change after the rule is activated.

Rule templates		
$w_0\ w_1$,	$w_0\ w_{-1}$,	$w_0\ t_1$
$w_0\ t_{-1}$,	$w_0\ c_1$,	$w_0\ c_{-1}$

Table 6.2 The rule templates used in bigram rule generation for capitalisation generation (w: words; t: NE types; c: capitalisation types). Subscripts define the distance from the current word

Particular importance must be given to the effect of words encountered in the test data which have not been seen in the training data. One way of improving the situation is to build separate rules for unknown words. The training data are divided into two groups. If words in one group are not seen in the other group, these words are regarded as unknown words. The same rule generation procedures are then applied.

The capitalisation generation system proposed in this section consists of 8 steps. These steps are depicted in Figure 6.2. Word sequences with NE classes and punctuation marks are processed by these 8 steps.

The first four steps in Figure 6.2 are straightforward processes. In step 1, the first character of the first word in each sentence is capitalised. Then in step 2, the first characters of NE words whose NE classes are 'ORGANIZATION', 'PERSON' or 'LOCATION' are capitalised. In step 3, initial words (e.g. B. B. C.) are capitalised, but only the first character is capitalised if the length of the initial word is longer than one character (e.g. Mr.). The word 'i' is treated differently, because this word normally means 'me' and is capitalised in this case. In step 4, backchannels (e.g. uhhuh) are de-capitalised.

As already mentioned in Table 2.3, 19.55% of NE words are not capitalised. Among non-NE words which are not first words in sentences, 2.32% of words are capitalised (e.g. El Nino). In order to dis-ambiguate capitalisation types, a frequency table of words which contains counts of words based on NE classes are looked-up. This frequency table is constructed on DB98, because DB98 is the only training data which is provided with reference NE classes.

Steps 5, 6, and 7 are related to the frequency table look-up. In step 5, the most frequent capitalisation type within NE classes is given to NE words which are not first words in sentences. In step 6, the same process is applied to non-NE words which are not first words in sentences. In step 7, if a word with the 'ORGANIZATION' class is a first word in a sentence, and its most frequent capitalisation type is All_Cap, then the capitalisation type of this word is changed to All_Cap.

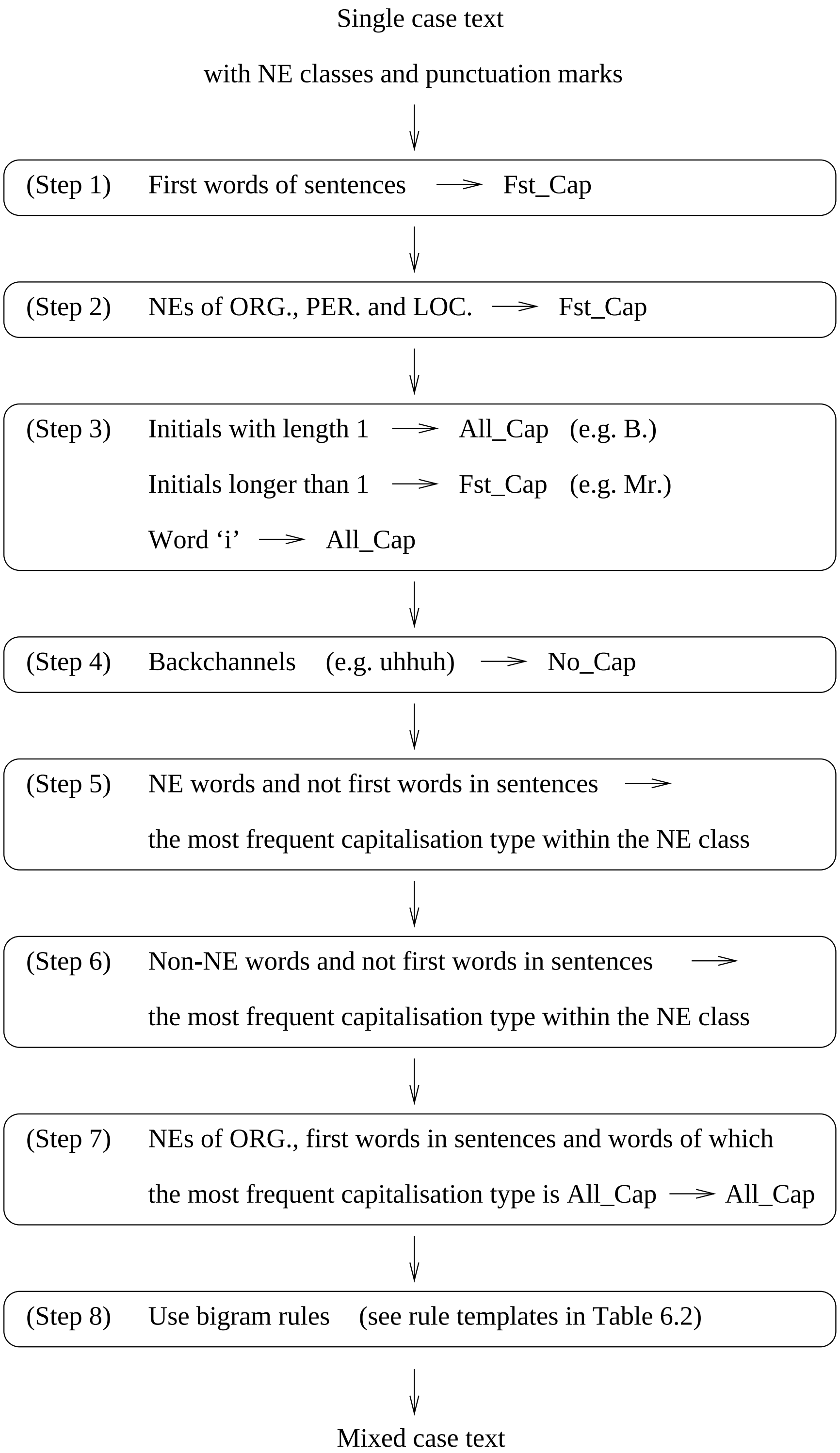

Figure 6.2 Procedures of the capitalisation generation system based on NE recognition and punctuation generation

In order to dis-ambiguate the capitalisation type of words which have more than one capitalisation type, the bigram rules generated from 6 rule templates described in Table 6.2 are applied one-by-one in step 8 according to a given order. If the conditions for a rule are met, then the rule is triggered and the classification type of the words is changed if necessary.

6.2 Experiments

There are two different systems of generating capitalisation: a system modified from a speech recogniser (described in Section 6.1.1) and a system based on NE recognition and punctuation generation (described in Section 6.1.2). These systems are summarised in Table 6.3.

System	Description
S_fr_SR	System modified from a speech recogniser
S_on_NE_P	System based on NE recognition and punctuation generation

Table 6.3 Description of automatic capitalisation generation systems

These systems cover the three types of capitalisation: all characters of a word are capitalised (All_Cap), only first character of a word is capitalised (Fst_Cap), and every character of a word is de-capitalised (No_Cap). The categories of capitalisation types were described in Table 1.1.

The results of both systems are compared on the basis that every procedure is fully automated. Then, the performance of the system based on NE recognition and punctuation generation is investigated with additional information: reference word sequences, reference NE classes and reference punctuation marks. As this system follows the 8 steps described in Figure 6.2, the effect of each step is examined when reference word sequences, reference NE classes, and reference punctuation marks are provided.

As described in Section 3.2.1, the performance of an automatic capitalisation generation system can be measured by the version 0.7 of the NIST Hub-4 IE scoring pipeline package. In the mixed case output, the words whose capitalisation types are All_Cap are surrounded by the "ORGANIZATION" NE class starting and end tags, and the words whose types are Fst_Cap by the "PERSON" NE class tags. Then, the words in the output are changed into single case. The same modification is applied to the reference text. Then the scoring pipeline package proceeds with these modified texts. TDB98 is used as test data.

6.2.1 Results: The system modified from a speech recogniser

The first automatic capitalisation system is implemented by slight modifications of the HTK Broadcast News (BN) transcription system. The HTK system was mentioned in Section 5.2.3, and details about the development of the HTK BN transcription system are given in [89].

First, every word in the pronunciation dictionary of the HTK system is duplicated with its pronunciation into three different capitalisation types (All_Cap, Fst_Cap, and No_Cap). Second, its language model is re-trained on mixed case transcriptions of BNtext92_97 and DB98.

Table 6.4 shows the results of capitalisation generation for TDB98 using this system. The performance of the system is measured by WER. When WER is measured, words are changed into single case from reference and hypothesis in order to measure the pure speech recognition rate. As the speech recognition output contains punctuation marks, WER$''$ which is the WER after punctuation marks are removed and words are changed to single case from reference and hypothesis is introduced. A similar concept was introduced as WER$'$ in punctuation generation in Section 5.2.3. WER$'$ was defined as the WER after punctuation is removed in reference and hypothesis.

System	WER	WER$''$	Precision	Recall	F-measure	SER
S_fr_SR	22.97	17.27	0.7736	0.6942	0.7317	48.55

Table 6.4 Results of capitalisation generation for TDB98 using the system modified from the HTK system. (WER: Word Error Rate (%); WER$''$: WER after punctuation is removed; SER: Slot Error Rate (%))

For punctuation generation, the HTK system reported 22.73% of WER and 17.04% of WER$'$ in Section 5.2.3. The difference between WER in punctuation generation and that in capitalisation generation is measured as 0.24%, and the difference between WER$'$ and WER$''$ is measured as 0.23%. These degradations are caused by the introduction of increased size of vocabulary and pronunciation dictionary. Two problems caused by this introduction were discussed in Section 6.1.1 The performance degradations are analysed as follows:

1. Distortion of LM

   In many cases, first words in sentences are non-NEs. Most of these words are not capitalised, if they are used in the middle of sentences. As there are 1,873 sentences in TDB98, the average number of words in a sentence in TDB is 16.9 words. Among the first words in sentences, 91.3% of these words are not NEs. Therefore, approximately, 5.4% ($(1/16.9) \times 0.913$) of counted word sequences are wrong, because a capitalised word and a de-capitalised word should be regarded as different words even if they have the same character sequence.

2. Sparser LM

   As the size of vocabulary is increased, LMs are sparser and estimating probabilities of word sequences becomes more difficult. The HTK system generates initial hypotheses using trigram language models and re-scores these hypotheses using 4-gram language models. As the size of vocabulary is multiplied by three, these LMs are sparser and the search space is widened.

If capitalisation generation is performed for a single case speech recogniser output as described in Section 6.1.2, mixed case output can be obtained without any loss in WER of speech recognition.

F-measure, precision, and recall are measured for this system as 0.7317, 0.7736, and 0.6942 respectively. The SER is measured as 48.55%. In addition to the effects for capitalisation generation, caused by the two factors of speech recognition degradation, loss of half scores in the evaluation of capitalisation generation affects the performance. If NE recognition and capitalisation generation are performed as post-processing of speech recognition, it is possible to obtain half scores for the words which are mis-recognised in speech recognition but are located next to NE signalling words.

6.2.2 Results: System based on NE recognition and punctuation generation

The steps of the capitalisation generation system depicted in Figure 6.2 start from the single case speech recognition output with punctuation marks and NE classes. In this system, multiple hypotheses which include punctuation marks are produced by the HTK system and are re-scored by prosodic information. Then NE recognition is performed for this speech recognition output. Capitalisation generation follows this speech recognition output with generated NE classes.

The results of automatic punctuation generation according to various scale factors to the prosodic feature model were presented in Table 5.11. The scale factor to prosodic feature model is set to be 0.71 at which WER is minimised. In this case, the WER and WER$'$ are measured as 22.55% and 16.86% for TDB98 respectively. Table 6.5 summarises the conditions and results of the automatic punctuation generation system used in this capitalisation generation system. Further details of this prosody combined system for punctuation generation and speech recognition were given in Section 5.1.2.

Punctuation generation system used	WER	WER′	Precision	Recall	F-measure	SER
S_H100 (α=0.71)	22.55	16.86	0.6088	0.3252	0.4239	83.36

Table 6.5 Summary of the punctuation generation system used in the capitalisation generation system (S_on_NE_P). Results are measured for TDB98. (α: scale factor to prosodic feature model; WER: Word Error Rate (%); WER′: WER after punctuation is removed; SER: Slot Error Rate (%))

NE recognition is performed for the best re-scored hypothesis. As an NE recogniser, the rule-based NE recogniser trained under the condition of 'with punctuation and name lists but without capitalisation' is used. This NE recogniser reported an F-measure of 0.9007 in Table 4.8 for the reference transcription of TDB98. Table 6.6 summarises conditions of the NE recogniser and its NE recognition performance for the reference transcription of TDB98. More details of this NE recogniser were discussed in Section 4.2.

Conditions of used NE recognition system	F-measure	SER(%)
Baseline+NL+Punc	0.9007	16.68

Table 6.6 Conditions of the rule-based NE recogniser used in the capitalisation generation system (S_on_NE_P) and its performance for the reference transcription of TDB98 (SER: Slot Error Rate)

The frequency table and bigram rules are constructed using the transcription of DB98. Table 6.7 shows the result of capitalisation generation based on NE recognition and punctuation generation. As this system does not increase the size of vocabulary, there is no degradation in WER and WER″. Compared to the other capitalisation generation system (S_fr_SR), this system (S_on_NE_P) shows better results by: 0.42% in WER, 0.41% in WER″, 2.62% in SER, and 0.0089 in F-measure. The factors which cause these differences were explained as 'distortion of LM', 'sparser LM', and 'loss of half scores' in Section 6.2.1. An example of capitalisation generation output produced by S_on_NE_P for a speech recognition result is shown in Figure 5 in the Appendix.

System	Test condition			Result					
	Word	NE	Punc.	WER	WER″	Precision	Recall	F-measure	SER
S_on_NE_P	Gen.	Gen.	Gen.	22.55	16.86	0.8094	0.6826	0.7406	45.93

Table 6.7 Results of the capitalisation generation system based on NE recognition and punctuation generation. (Punc.: Punctuation; Gen.: Generated; WER: Word Error Rate (%); WER″: WER after punctuation is removed; SER: Slot Error Rate (%))

6.3 Analysis of performance of the system based on NE recognition and punctuation generation

The effects of speech recognition errors, NE recognition errors and punctuation generation errors are accumulated in the results of S_on_NE_P in Table 6.7. In this section, the performance of S_on_NE_P is investigated by including one or more of the following: reference word sequences, reference NE classes and reference punctuation marks. The total effects of the accumulated errors are examined, and the contribution of each step in S_on_NE_P is tested for reference word sequences, NE classes and punctuation marks. Then, the effects of speech recognition and punctuation generation errors are examined. The performance of S_on_NE_P is compared with that of Microsoft Word 2000.

6.3.1 The contribution of each experimental step

In order to measure the pure contribution of each step in the capitalisation generation system based on NE classes and punctuation marks, the contribution of each step is examined for reference word sequence, reference NE classes and reference punctuation marks.

Table 6.8 shows the result of the capitalisation generation system based on NE classes and punctuation marks for these test conditions. The F-measure is measured as 0.9756 and the SER as 4.89%. After removing the effects of speech recognition errors, NE recognition errors and punctuation generation errors, the F-measure is improved by 0.2350 (0.9756 - 0.7406) and the SER by 41.04% (45.93 - 4.89).

	Test condition			Result			
System	Word	NE	Punc.	Precision	Recall	F-measure	SER
S_on_NE_P	Ref.	Ref.	Ref.	0.9726	0.9786	0.9756	4.89

Table 6.8 Results of the capitalisation generation system based on NE classes and punctuation marks for reference word sequences, NE classes and punctuation marks. (Punc.: Punctuation; Ref.: Reference)

Table 6.9 shows the capitalisation generation results with different combinations of experimental steps. By just performing step 1 (the first character of the first word in each sentence is capitalised), the F-measure of 0.5494 is already obtained, although the recall (0.3814) is quite poor to the precision (0.9818). By performing step 2, in addition to step 1, the F-measure is increased to 0.8448.

With steps 1, 2, 3 and 4, which can be done by straightforward processes without the need for training data, an F-measure of 0.9247 is obtained for capitalisation generation. With steps 5, 6 and 7 which depend on the use of frequency tables, the result can be increased to 0.9694. In addition, 0.9756 points in F-measure are achieved using bigram rules. Table 6.9 shows these results.

Included step								Result			
1	2	3	4	5	6	7	8	Precision	Recall	F-measure	SER(%)
I								0.9818	0.3814	0.5494	62.57
I	I							0.8944	0.8004	0.8448	29.41
I	I	I						0.9581	0.8881	0.9218	15.08
I	I	I	I					0.9632	0.8881	0.9241	14.58
I	I	I	I	I				0.9817	0.9019	0.9401	11.45
I	I	I	I	I	I			0.9703	0.9681	0.9692	6.16
I	I	I	I	I	I	I		0.9705	0.9683	0.9694	6.12
I	I	I	I	I	I	I	I	0.9726	0.9786	0.9756	4.89

Table 6.9 Results of capitalisation generation with different combinations of processing steps

6.3.1.1 Analysis: The result of capitalisation generation when reference word sequences, NE classes and punctuation marks are provided

The capitalisation generation system based on NE classes and punctuation marks reports an F-measure of 0.9756 with 236 errors for TDB98 when reference word sequences, punctuation marks and NE classes are provided. These 236 errors can be categorised into the following three groups:

1. Errors due to the inconsistency of capitalisation (Group 1)
2. Errors due to limited number of observations in training data (Group 2)
3. Errors not included in Group 1 and Group 2 (Group 3)

Groups 1 and 2 are not totally exclusive of each other. The number of errors in Group 1 can be measured by substituting the training data with the test data and repeating the experiment. After this substitution, there were still 100 errors with an F-measure of 0.9896. These 100 errors were examined manually. Most of them are caused by inconsistency of capitalisation which cannot be corrected by bigrams. For example:

- News in “Lisa Stark, A. B. C. News, Washington” (normally A. B. C. news)
- the President (normally the president apart from the President of U. S. A.)
- World Today (programme name)
- South, East .... (normally south, east but sometimes capitalised in weather forecast)
- Main Street in “U. S. props up Japan’s currency from Wall Street to Main Street” (normally main street)

The errors in Group 2 show that they can be corrected if the size of the training data is increased. Assume that a word in test data is observed enough if it is observed in training data more than twice (≥ 3) with its NE class and its capitalisation type. On this assumption, capitalisation errors in Group 2 can be categorised into the following 4 sub-categories:

1. Errors at an unknown word (Group 2-1)
2. Errors at a word never seen in the training data with its NE class (Group 2-2)
3. Errors at a word seen only once in the training data with its NE class (Group 2-3)
4. Errors at a word seen twice in the training data with its NE class (Group 2-4)

Among 236 total errors, the number of errors in Group 2-1, 2-2, 2-3 and 2-4 are counted as 25, 23, 9 and 0 respectively. These numbers constitute 24.15% of total errors.

Errors in Group 3 illustrate the fact that the training data cannot reflect the test data perfectly, because a word which has a capitalisation type error in this group is observed enough with its NE class. As these errors are not caused by the inconsistency of capitalisation, the correct response for these errors is limited for the current methodology of capitalisation generation.

Among these three categories of errors in capitalisation generation, only the errors in Group 2 can be corrected if the size of the training data is increased. The errors in Group 2 consist of 25.85% of total errors and the F-measure of the system on the current input condition is 0.9756. If the errors in Group 2 are corrected, the F-measure of this capitalisation generation system is expected to be increased to:

$$0.9756 + (1 - 0.9756) \times 0.2585 = 0.9819 \tag{6.1}$$

At the moment, it is believed that the result of an F-measure of 0.9756 in capitalisation generation on the condition of reference word sequences, punctuation marks and NE classes is a good result given the relatively small amount of training data.

6.3.2 The effect of NE recognition errors

In order to measure the effect of NE recognition errors in the capitalisation generation system based on NE classes and punctuation marks, the results of capitalisation generation are examined for reference word sequences and reference punctuation marks. However, NE classes are generated by an NE recogniser. As an NE recogniser, the rule-based NE recogniser trained under the condition of 'with punctuation and name lists but without capitalisation' is used. It recognises NEs with 0.9007 in F-measure and 16.68% in SER for TDB98. Table 6.6 summarised conditions of the NE recogniser and its performance for NE recognition.

Table 6.10 shows the results of capitalisation generation for reference word sequences, generated NE classes and reference punctuation marks. As the F-measure of capitalisation generation for reference word sequences, NE classes and punctuation marks was measured as 0.9756, the effect of NE recognition errors on capitalisation generation is measured with a degradation in F-measure of 0.0158 (0.9756 - 0.9585). The degradation in SER is measured as 3.20%.

System	Test condition			Result			
	Word	NE	Punc.	Precision	Recall	F-measure	SER
S_on_NE_P	Ref.	Gen.	Ref.	0.9552	0.9643	0.9598	8.09

Table 6.10 Results of capitalisation generation for reference word sequences, generated NE classes and reference punctuation marks. (Punc.: Punctuation; Ref.: Reference; Gen.: Generated)

6.3.2.1 Analysis: the effect of NE recognition errors

Steps 2, 5, 6 and 7 of the capitalisation generation system described in Figure 6.2 are based on NE classes. In this section, the effect of NE recognition errors for the overall performance of capitalisation generation is analysed.

The statistics of TDB98 were shown in Tables 3.10 and 3.11. According to these tables, the number of initial words which are NEs is 543 and the number of NE words which are first words in sentences and which have a capitalised first character is 143. Among NE words, these 543 initials and 143 NEs at the beginning of sentences can be capitalised correctly without the help of the NE recognition system. As the total number of NEs in TDB98 is 3,149, the number of NEs which require the help of the NE recognition system is roughly 2,463 (3,149 - 543 - 143).

As the F-measure of the used NE recogniser is 0.9007 for NE recognition, the capitalisation of about 245 (2,463 $\times$ (1 - 0.9007)) NE words may be affected by the NE recognition errors. This number of words constitutes 5.1% of total capitalised words. However, the actual degradation caused by the errors of NE recognition is measured as 0.0158. This implies that this capitalisation generation system is robust to NE recognition errors.

6.3.3 The effect of punctuation generation errors

In order to measure the effect of punctuation generation errors in the capitalisation generation system based on NE classes and punctuation marks, the results of capitalisation generation are examined for reference word sequences, reference NE classes and generated punctuation marks. The punctuation generation system using combined information of an LM and a prosodic feature model is used. It generates punctuation marks with an F-measure of 0.7830 and an SER of 32.30% for the reference transcription of TDB98. Table 6.11 summarises this punctuation generation system. More details of this punctuation generation system were given in Section 5.1.

Used punctuation generation system	F-measure	SER(%)
S_LM+CART	0.7830	32.30

Table 6.11 The performance of punctuation generation for the reference transcription of TDB98 produced by the punctuation generation system using combined information of an LM and a prosodic feature model (SER: Slot Error Rate)

Table 6.12 shows the result of capitalisation generation for reference word sequences, reference NE classes and generated punctuation marks. As the F-measure of capitalisation generation for reference word sequences, NE classes and punctuation marks was measured as 0.9756, the effect of punctuation generation errors on capitalisation generation is measured as an F-measure of 0.0909 (0.9756 - 0.8847). The degradation in SER is measured as 18.21%.

System	Test condition			Result			
	Word	NE	Punc.	Precision	Recall	F-measure	SER
S_on_NE_P	Ref.	Ref.	Gen.	0.8832	0.8861	0.8847	23.10

Table 6.12 Results of capitalisation generation for reference word sequences, reference NE classes and generated punctuation marks. (Punc.: Punctuation; Ref.: Reference; Gen.: Generated)

6.3.3.1 Analysis: The effect of punctuation generation errors

Steps 1, 5, 6 and 7 of the capitalisation generation system depicted in Figure 6.2 are based on punctuation marks. According to the statistics of TDB98 shown in Tables 3.10 and 3.11, the number of non-NE words which have a capitalised first character and which are first words in sentences is 1,603.

Punctuation marks whose place is correct but type is wrong are meaningful in punctuation generation and obtain half scores. However, punctuation type errors between commas and full stops, and between commas and question marks are not meaningful for capitalisation generation, because the words next to commas are normally de-capitalised. If the half scores are given in punctuation generation only between full stops and question marks, the F-measure of punctuation generation decreases to 0.6826.

The maximum number of words whose capitalisation types are possibly affected by punctuation generation errors can be roughly estimated as 1,603 $\times$ (1 - 0.6826) = 509. This number of words constitute 10.56% of the total number of capitalised words. The actual degradation caused by punctuation generation errors is measured as an F-measure of 0.0909. This implies that most punctuation generation errors cause errors in capitalisation generation, but the number of errors caused in capitalisation generation do not exceed the number of errors in punctuation generation.

6.3.4 The correlation between the effects of NE recognition errors and the effects of punctuation generation errors

In this section, the correlation between the effects of NE recognition errors and those of punctuation generation errors to capitalisation generation are examined. NE recognition and punctuation generation are performed for the reference transcription of TDB98, in which every word is de-capitalised and every punctuation mark is removed. The rule-based NE recogniser and the punctuation generation system, which uses the combined information of an LM and a prosodic feature model, are used.

Using these NE recogniser and punctuation generation systems, punctuation marks are produced first for the transcription of TDB98, then NE recognition is performed for the reference transcription with these generated punctuation marks. The capitalisation generation is carried out for this result of NE recognition and punctuation generation for the transcription of TDB98.

Table 6.13 shows the results of capitalisation generation for NE recognition and punctuation generation output from reference word sequences. The simultaneous effects of NE recognition errors and punctuation generation errors on capitalisation generation are measured as a degradation in F-measure of 0.1065 and in SER of 21.36%. As the effect of NE recognition errors on capitalisation generation and the effect of punctuation generation errors on capitalisation generation are measured as 0.0158 and 0.0909 in F-measure respectively (3.20% and 18.21% in SER respectively), it is shown that these simultaneous effects are almost equivalent to the sum of individual effects. This suggests that the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation.

System	Test condition			Result			
	Word	NE	Punc.	Precision	Recall	F-measure	SER
S_on_NE_P	Ref.	Gen.	Gen.	0.8667	0.8715	0.8691	26.25

Table 6.13 Results of capitalisation generation for reference word sequences, generated NE classes, and generated punctuation marks (Punc.: Punctuation; Ref.: Reference; Gen.: Generated)

6.3.5 Comparison with Microsoft Word 2000

The results of automatic capitalisation generation using Microsoft Word 2000 were reported in Table 6.1 for the first 10.7% words of TDB98. In this section, the performance of S_on_NE_P is compared with that of Microsoft 2000 for the same part of TDB98. As the reference sequence of words and punctuation marks were given as input when automatic capitalisation generation was performed by Microsoft Word 2000, capitalisation is generated by S_on_NE_P for the reference word sequences, generated NE classes and reference punctuation marks. Table 6.14 shows the results of capitalisation generation by S_on_NE_P for the first 10.7% words of TDB98. Compared to Microsoft, S_on_NE_P shows better results by 0.0687 in F-measure and by 11.62% in SER.

System	Test condition			Result			
	Word	NE	Punc.	Precision	Recall	F-measure	SER
S_on_NE_P	Ref.	Gen.	Ref.	0.9588	0.9608	0.9598	8.04
MS Word 2000	Ref.	N/A	Ref.	0.9987	0.8045	0.8911	19.66

Table 6.14 Results of capitalisation generation by S_on_NE_P for reference word sequences, generated NE classes and reference punctuation marks using 10.7% of TDB98. These results are compared with those from Microsoft Word for the same part of TDB98. (Punc.: Punctuation; Ref.: Reference; Gen.: Generated)

6.3.6 Estimation: Results of the system based on NE recognition and punctuation generation when every procedure is fully automated

In Section 6.2.2, the capitalisation generation system based on NE recognition and punctuation generation reported an F-measure of 0.7406. In this section, this result is compared with the results expected from the previous conclusions: the performance of NE recognition is degraded linearly according to speech recognition errors (Section 4.3.4), and the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation (Section 6.3.4).

The experiment in Section 6.2.2 used a punctuation generation system which reported an F-measure of 0.4239 at a scale factor of 0.71 and reported 16.86% of WER′ (WER after removing punctuation marks from a reference and a hypothesis) at this scale factor. In addition to this punctuation generation system, the experiment used an NE recognition system which reported an F-measure of 0.9007. Since an experiment in Section 4.3.4 reported that the performance of an NE recogniser is linearly degraded by 0.0062 points in F-measure per 1% of additional WER, the capitalisation generation system based on NE recognition and punctuation generation is expected to obtain the following F-measure for NE recognition:

$$0.9007 - 0.0062 \times 16.86 = 0.7962 \tag{6.2}$$

As shown in Section 6.3.2, the result of capitalisation generation is degraded by an F-measure of 0.0158 due to NE recognition error of an F-measure of 0.0993 (1 - 0.9007). The degradation of capitalisation generation caused by NE recognition errors (assuming that this degradation is proportional to NE recognition errors) is expected to be:

$$0.0158 \times \frac{1 - 0.7962}{0.0993} = 0.0324 \tag{6.3}$$

As shown in Section 6.3.3, the result of capitalisation generation is degraded by an F-measure of 0.0909 due to punctuation generation errors of an F-measure of 0.2170 (1 - 0.7830). The degradation of capitalisation generation caused by punctuation generation errors (assuming that this degradation is proportional to punctuation generation errors) is expected to be:

$$0.0909 \times \frac{1 - 0.4292}{0.2170} = 0.2391 \tag{6.4}$$

If it is assumed that the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation, the total degradation of capitalisation generation caused by NE recognition errors and punctuation generation errors is expected to be:

$$0.9756 - 0.0324 - 0.2391 = 0.7041 \tag{6.5}$$

Based on this expectation, the result of capitalisation generation of an F-measure of 0.7406 is believed to be a reasonable result when every procedure is fully automated.

6.4 Summary

In this chapter, another important area of transcription readability improvement, automatic capitalisation generation, has been discussed. Two different systems have been proposed for this task. The first is a slightly modified speech recogniser. In this system, every word in its vocabulary is duplicated: one is given in a de-capitalised form and the others are in capitalised forms. In addition, its language model is re-trained on mixed case texts. The other system is based on NE recognition and punctuation generation since most capitalised words are first words in sentences or NE words.

In order to compare the performance of the proposed systems, experiments of automatic capitalisation generation were performed for TDB98. The results of both systems have been compared on the basis that every procedure is fully automated. The system based on NE recognition and punctuation generation showed better results in WER, in F-measure and in SER than the system modified from the speech recogniser, because the latter system has distortion of LM, sparser LM, and loss of half scores.

The system based on NE recognition and punctuation generation follows the 8 steps described in Figure 6.2. The effect of each step was examined when reference word sequences, reference NE classes, and reference punctuation marks are provided. More than 0.92 points in F-measure of capitalisation has been generated by straightforward steps without the need for training data.

The performance of the system based on NE recognition and punctuation generation has been investigated for the additional clues: reference word sequences, reference NE classes and reference punctuation marks. The results showed that this system is robust to NE recognition errors and that the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation.

Chapter 7
Conclusions and further work

In this chapter, a review of the work is given, highlighting the contributions and important results. The thesis concludes with some proposals for future research.

7.1 Review of the contributions of this thesis

In this thesis, a rule-based Named Entity (NE) recognition system which generates rules automatically has been devised and an automatic punctuation generation system using prosodic information has been proposed. An automatic capitalisation generation system has been designed using the NE recognition system and the punctuation generation system.

Previous work regarding the NE task were mainly categorised by hand-crafted rule-based systems and stochastic systems. In Chapter 4, an automatic rule generating method, which uses the Brill rule inference approach, was proposed for the NE task. For automatic punctuation generation, the previous work assumed that sentence boundaries are pre-determined or that the input speech comes from a very small number of speakers. In Chapter 5, a complete automatic punctuation generation method consisting of a speech recogniser with a few straightforward modifications. Further improvement in punctuation generation was achieved by re-scoring multiple hypotheses using prosodic information. The fact that most capitalised words are first words in sentences or NE words motivated a capitalisation generation method based on NEs and sentence boundaries. In Chapter 6, an automatic means of capitalisation generation based on NE recognition and punctuation generation was discussed.

7.1.1 Rule-based Named Entity (NE) recognition

In order to measure the performance of the rule-based NE recognition system, it was compared with that of IdentiFinder, BBN's HMM-based system which gave the best performance among the systems that participated in the 1998 Hub-4 benchmark test. For the baseline case (with no punctuation, no capitalisation, and no name list), both systems showed almost equal performance and did likewise in the case of additional information such as punctuation, capitalisation and name lists. When input texts were corrupted by speech recognition errors, the performance of both systems were degraded linearly with increasing WER at almost the same rate. Although this rule-based approach is different from the stochastic method, which is recognised as one of the most successful methods, this rule-based system gave the same level of performance.

7.1.2 Automatic punctuation generation

The proposed punctuation generation system incorporated prosodic information with acoustic and language model information. Experiments were conducted first for the reference transcriptions. In these experiments, prosodic information was shown to be more useful than language model information. When these information sources are combined, an F-measure of up to 0.7830 was obtained for punctuation generation of a reference transcription.

A few straightforward modifications of a conventional speech recogniser allowed the system to produce punctuation marks and speech recognition hypotheses simultaneously. The multiple hypotheses were produced by the automatic speech recogniser and were re-scored by prosodic information. When prosodic information is incorporated, the F-measure was improved by 19% relative. At the same time, small reductions in word error rate were obtained.

7.1.3 Automatic capitalisation generation

Two different systems were proposed for this task. The first system is a slightly modified speech recogniser. In this system, every word in its vocabulary is duplicated: one in a de-capitalised form and the others in capitalised forms. In addition, its language model is re-trained on mixed case texts. The other system is based on NE recognition and punctuation generation, since most capitalised words are first words in sentences or NE words.

Both systems were compared first on the condition that every procedure is fully automated. The system based on NE recognition and punctuation generation showed better results in word error rate, in F-measure and in SER than the system modified from a speech recogniser, because the former system does not have the distortions of the LM, a sparser LM, and loss of half scores.

The performance of the system based on NE recognition and punctuation generation was investigated by including one or more of the following: reference word sequences, reference NE classes and reference punctuation marks. The results showed that this system is robust to NE recognition errors. Although most punctuation generation errors cause errors in this capitalisation generation system, the number of errors caused in capitalisation generation does not exceed the number of errors in punctuation generation. In addition, it showed that the effect of NE recognition errors is independent of the effect of punctuation generation errors for capitalisation generation.

7.2 Suggested further work

The examination of NE recognition, punctuation generation and capitalisation generation has been conducted in this thesis. If long distance lexical information and POS information had been incorporated, then the performance of the systems would have been improved considerably in decisions about exact boundaries of NEs and sentences. Any further work must include a methodology which improves the performance of the NE recognition system, the punctuation generation system, and the capitalisation generation system using syntactic information, and a methodology which generates a sufficient number of punctuation marks using more precise design for the pronunciation of punctuation marks. In addition to these, a new task definition of NE recognition stimulates more precise extraction of numeric entities.

7.2.1 The use of syntactic information

Syntactic structure information concerns how words can be put together, and determines what structural role each word plays and which phrases are subparts of which other phrases [18, 36, 37]. Some words are left-attached and others are right-attached. In addition, the same words can be used differently according to their syntactic functions. The current systems do not consider this information source.

A possible solution for this improvement is parsing. Using parsing results, rules can be generated according to the relationship between head words of a parent node and words of a child node. Complete parsing of sentences is very difficult for unrestricted input text. In addition, when input text is derived from speech, due to corruption by speech recogniser error and missing punctuation, complete parsing is almost impossible [17]. However, some syntactic fragments such as noun groups and verb groups are identified relatively reliably, and are very useful when deciding NE boundaries and sentence boundaries.

Prosody information such as pitch, duration and energy gives clues when identifying sentence structure [39, 43, 84]. In speech recognition, the use of prosody is limited because prosodic information in an utterance does not help significantly with the low level identification of words. The patterns of changing pitch in the voice over an utterance plays a role in guiding the prosodic structure of the utterance. Further studies are needed on utilising prosodic information to improve the understanding of syntactic structure.

7.2.2 More precise definition of pronunciation for punctuation marks

A few straightforward modifications of a conventional speech recogniser allowed the system to produce punctuation marks and speech recognition hypotheses simultaneously. This system generated punctuation marks of an F-measure of 0.4400 with 0.5811 of precision and 0.3541 of recall. There is a big difference between the values of precision and recall. Compared to the punctuation generation result for the reference transcription, this precision is adequate, but the recall is too low. This showed that insufficient punctuation marks are generated in the hypotheses.

One of the modifications to the speech recogniser is that the pronunciations of punctuation marks are registered as silence. This is only a rough approximation. About 24% of punctuation marks are not related to silence in broadcast news. In order to improve the result of punctuation generation, a more precise definition of the pronunciation for punctuation marks is needed.

An alternative approach for generating punctuation marks from the 1-best speech recogniser output which does not have punctuation marks has been proposed in this thesis. An extension of this approach is to generate punctuation marks from N-best speech recogniser output which does not have punctuation marks, and re-score these N-best output using the prosodic feature model. This may produce improved results without assuming that acoustic pronunciation of punctuation is silence.

7.2.3 New NE task definition

A new task definition (for version 1.4, see [14]) was proposed to include more NE classes such as:

- DURATION: a measurement of time elapsed or period of time during which something lasts
- MEASURE: standard numeric measurement phrases such as age, area, distance, energy, speed, temperature, volume and weight.
- CARDINAL: a numerical count or quantity of some object (in the form of numbers, decimals or fractions)

Since these additional NE classes are related to numeric expressions, it is clear that more importance should be given to numeric expressions. At this time, there is a difference of about 3.8 points in F-measure between IdentiFinder (0.8777) and the rule-based NE recognition system (0.8398) for numeric entities. This is small from the overall view since the numeric entities account for about 7.5 percent of the total number of NEs. However in the new task definition, numeric entities are becoming more important. New rule templates or regular rules for numeric entities need to be developed.

Appendix

In the appendix, examples of a reference text and hypothesis texts are shown as follows:

1. Example of a reference text (Figure 1): First 340 words of the TDB98 reference transcription is shown in mixed case with NE tags and punctuation marks.

2. Example of an NE recognition output (Figure 2): An NE recognition output is generated by the rule-based NE recognition system using name lists for the same part of the TDB98 reference transcription. Punctuation marks are provided, but capitalisation information is not. In this condition, the rule-based NE recognition system reported an F-measure of 0.9007 and an SER of 16.68% as shown in Table 4.8.

3. Example of a punctuation generation output (Figure 3): A punctuation generation output is produced for the same part of the TDB98 reference transcription (in single case) by the combined system of a language model and a prosodic feature model (S_LM+CART). In this condition, S_LM+CART reported an F-measure of 0.7830 and an SER of 32.30% as shown in Table 5.6.

4. Example of a capitalisation generation output (Figure 4): A capitalisation generation output is produced for the same part of the TDB98 reference transcription by the capitalisation system based on NE recognition and punctuation generation (S_on_NE_P). NE recognition is performed by the rule-based NE recognition system. Punctuation marks are generated by S_LM+CART. In this condition, S_on_NE_P reported an F-measure of 0.8691 and an SER of 26.25% as shown in Table 6.13.

5. Example of a capitalisation generation output for a speech recognition result (Figure 5): NE recognition is performed by the rule-based NE recognition system for the speech recognition results of the HTK system. This speech recognition output contains punctuation marks. Capitalisation generation is performed by S_on_NE_P. In this condition, S_on_NE_P reported an F-measure of 0.7406 and an SER of 45.93% as shown in Table 6.7.

The guardians of the electronic stock market <b_enamex TYPE="ORGANIZATION"> NASDAQ <e_enamex> who've been burned by past ethics questions, are moving to head off market fraud by toughening the rules for companies that want to be listed on the exchange. Marketplace's <b_enamex TYPE="PERSON"> Philip Boroff <e_enamex> reports. As part of the proposals, penny stocks will be eliminated from <b_enamex TYPE="ORGANIZATION"> NASDAQ <e_enamex>. These trade for literally <b_numex TYPE="MONEY"> pennies <e_numex>. Less than <b_numex TYPE="MONEY"> a dollar <e_numex> a share. They're the stocks of speculative companies. On wall street, they're the longest of the long shots. Some penny stocks grow into established corporations. Others are shell companies. Incorporated firms without assets or prospects. Some of these are sold by small unsavory brokerage firms. That dump them upon gullible investors. <b_enamex TYPE="PERSON"> David Whitcomb <e_enamex> is a <b_enamex TYPE="ORGANIZATION"> Rutgers University <e_enamex> finance professor and frequent <b_enamex TYPE="ORGANIZATION"> NASDAQ <e_enamex> critic. That's the real change, it's reducing the status of cheap stocks so that at least <b_enamex TYPE="ORGANIZATION"> NASDAQ <e_enamex> is not giving them its seal of approval. Also, these companies will no longer appear in newspapers on <b_enamex TYPE="ORGANIZATION"> NASDAQ <e_enamex>'s list. And <b_enamex TYPE="PERSON"> Whitcomb <e_enamex> says investors may be less prone to buy them if they're not listed in the paper. <b_enamex TYPE="ORGANIZATION"> NASDAQ <e_enamex> officials say, they're not only trying to fight fraud by raising listing standards, they're doing a periodic tuneup of their market. Which they hope will help promote public confidence. In <b_enamex TYPE="LOCATION"> New York <e_enamex>, I'm <b_enamex TYPE="PERSON"> Philip Boroff <e_enamex> for Marketplace. And that's the top of our news for <b_timex TYPE="DATE"> Thursday, November fourteenth <e_timex>. Today the <b_enamex TYPE="ORGANIZATION"> Dow Jones <e_enamex> industrial average gained thirty eight and three quarter points. Details when we do the numbers. Later on tonight's program, life in the fast lane. And coming up next, a fast food Godzilla joins the burger wars in <b_enamex TYPE="LOCATION"> Japan <e_enamex>. I'm <b_enamex TYPE="PERSON"> David Brancaccio <e_enamex>, this is Marketplace. At the foreign desk in <b_enamex TYPE="LOCATION"> San Francisco <e_enamex>, I'm <b_enamex TYPE="PERSON"> George Lewinski <e_enamex>. American popular culture whether it's rock and roll, fashion, or <b_enamex TYPE="LOCATION"> Hollywood <e_enamex> movies, has long been an important export. Even though statisticians have a hard time measuring its value. Take fast food. When the first American style burger joint opened in <b_enamex TYPE="LOCATION"> London <e_enamex>'s fashionable <b_enamex TYPE="LOCATION"> Regent street <e_enamex> some twenty years ago, it was mobbed. Now it's <b_enamex TYPE="LOCATION"> Asia <e_enamex>'s turn

Figure 1 Example of a reference text. The first 340 words of the TDB98 reference transcription in mixed case with NE tags and punctuation marks.

THE GUARDIANS OF THE ELECTRONIC STOCK MARKET <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> WHO’VE BEEN BURNED BY PAST ETHICS QUESTIONS ARE MOVING TO HEAD OFF MARKET FRAUD BY TOUGHENING THE RULES FOR COMPANIES THAT WANT TO BE LISTED ON THE EXCHANGE MARKETPLACE’S <b_enamex TYPE=“PERSON”> PHILIP BOROFF <e_enamex> REPORTS AS PART OF THE PROPOSALS PENNY STOCKS WILL BE ELIMINATED FROM <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> THESE TRADE FOR LITERALLY PENNIES LESS THAN <b_numex TYPE=“MONEY”> A DOLLAR <e_numex> A SHARE THEY’RE THE STOCKS OF SPECULATIVE COMPANIES ON WALL STREET THEY’RE THE LONGEST OF THE LONG SHOTS SOME PENNY STOCKS GROW INTO ESTABLISHED CORPORATIONS OTHERS ARE SHELL COMPANIES INCORPORATED FIRMS WITHOUT ASSETS OR PROSPECTS SOME OF THESE ARE SOLD BY SMALL UNSAVORY BROKERAGE FIRMS THAT DUMP THEM UPON GULLIBLE INVESTORS <b_enamex TYPE=“PERSON”> DAVID WHITCOMB <e_enamex> IS A <b_enamex TYPE=“ORGANIZATION”> RUTGERS UNIVERSITY <e_enamex> FINANCE PROFESSOR AND FREQUENT <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> CRITIC THAT’S THE REAL CHANGE IT’S REDUCING THE STATUS OF CHEAP STOCKS SO THAT AT LEAST <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> IS NOT GIVING THEM ITS SEAL OF APPROVAL ALSO THESE COMPANIES WILL NO LONGER APPEAR IN NEWSPAPERS ON <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex>’S LIST AND <b_enamex TYPE=“PERSON”> WHITCOMB <e_enamex> SAYS INVESTORS MAY BE LESS PRONE TO BUY THEM IF THEY’RE NOT LISTED IN THE PAPER <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> OFFICIALS SAY THEY’RE NOT ONLY TRYING TO FIGHT FRAUD BY RAISING LISTING STANDARDS THEY’RE DOING A PERIODIC TUNEUP OF THEIR MARKET WHICH THEY HOPE WILL HELP PROMOTE PUBLIC CONFIDENCE IN <b_enamex TYPE=“LOCATION”> NEW YORK <e_enamex> I’M <b_enamex TYPE=“PERSON”> PHILIP BOROFF <e_enamex> FOR MARKETPLACE AND THAT’S THE TOP OF OUR NEWS FOR <b_timex TYPE=“DATE”> THURSDAY NOVEMBER FOURTEENTH <e_timex> TODAY THE <b_enamex TYPE=“ORGANIZATION”> DOW JONES <e_enamex> INDUSTRIAL AVERAGE GAINED THIRTY EIGHT AND THREE QUARTER POINTS DETAILS WHEN WE DO THE NUMBERS LATER ON TONIGHT’S PROGRAM LIFE IN THE FAST LANE AND COMING UP NEXT A FAST FOOD GODZILLA JOINS THE BURGER WARS IN <b_enamex TYPE=“LOCATION”> JAPAN <e_enamex> I’M <b_enamex TYPE=“PERSON”> DAVID BRANCACCIO <e_enamex> THIS IS MARKETPLACE AT THE FOREIGN DESK IN <b_enamex TYPE=“LOCATION”> SAN FRANCISCO <e_enamex> I’M <b_enamex TYPE=“PERSON”> GEORGE LEWINSKI <e_enamex> AMERICAN POPULAR CULTURE WHETHER IT’S ROCK AND ROLL FASHION OR <b_enamex TYPE=“LOCATION”> HOLLYWOOD <e_enamex> MOVIES HAS LONG BEEN AN IMPORTANT EXPORT EVEN THOUGH STATISTICIANS HAVE A HARD TIME MEASURING ITS VALUE TAKE FAST FOOD WHEN THE FIRST AMERICAN STYLE BURGER JOINT OPENED IN <b_enamex TYPE=“LOCATION”> LONDON <e_enamex>’S FASHIONABLE REGENT STREET SOME TWENTY YEARS AGO IT WAS MOBBED NOW IT’S <b_enamex TYPE=“LOCATION”> ASIA <e_enamex>’S TURN

Figure 2 Example of an NE recognition output. An NE recognition output is generated by the rule-based NE recognition system using name lists for the same part of the TDB98 reference transcription. Punctuation marks are provided, but capitalisation information is not. Underlined words show the positions of NE recognition errors.

THE GUARDIANS OF THE ELECTRONIC STOCK MARKET NASDAQ WHO'VE BEEN BURNED BY PAST ETHICS QUESTIONS, ARE MOVING TO HEAD OFF MARKET FRAUD BY TOUGHENING THE RULES FOR COMPANIES THAT WANT TO BE LISTED ON THE EXCHANGE. MARKETPLACE'S PHILIP BOROFF REPORTS. AS PART OF THE PROPOSALS(.) PENNY STOCKS WILL BE ELIMINATED FROM NASDAQ. THESE TRADE FOR LITERALLY PENNIES. LESS THAN A DOLLAR A SHARE. THEY'RE THE STOCKS OF SPECULATIVE COMPANIES. ON WALL STREET, THEY'RE THE LONGEST OF THE LONG SHOTS. SOME PENNY STOCKS GROW INTO ESTABLISHED CORPORATIONS(,) OTHERS ARE SHELL COMPANIES(,) INCORPORATED FIRMS WITHOUT ASSETS OR PROSPECTS. SOME OF THESE ARE SOLD BY SMALL UNSAVORY BROKERAGE FIRMS. THAT DUMP THEM UPON GULLIBLE INVESTORS. DAVID WHITCOMB IS A RUTGERS UNIVERSITY FINANCE PROFESSOR AND FREQUENT NASDAQ CRITIC(,) THAT'S THE REAL CHANGE(.) IT'S REDUCING THE STATUS OF CHEAP STOCKS{,} SO THAT AT LEAST NASDAQ IS NOT GIVING THEM ITS SEAL OF APPROVAL. ALSO, THESE COMPANIES WILL NO LONGER APPEAR IN NEWSPAPERS ON NASDAQ'S LIST. AND WHITCOMB SAYS INVESTORS MAY BE LESS PRONE TO BUY THEM{.} IF THEY'RE NOT LISTED IN THE PAPER. NASDAQ OFFICIALS SAY[] THEY'RE NOT ONLY TRYING TO FIGHT FRAUD BY RAISING LISTING STANDARDS, THEY'RE DOING A PERIODIC TUNEUP OF THEIR MARKET(,) WHICH THEY HOPE WILL HELP PROMOTE PUBLIC CONFIDENCE. IN NEW YORK(.) I'M PHILIP BOROFF FOR MARKETPLACE. AND THAT'S THE TOP OF OUR NEWS FOR THURSDAY, NOVEMBER FOURTEENTH. TODAY THE DOW JONES INDUSTRIAL AVERAGE GAINED THIRTY EIGHT AND THREE QUARTER POINTS. DETAILS{,} WHEN WE DO THE NUMBERS[] LATER ON TONIGHT'S PROGRAM, LIFE IN THE FAST LANE. AND COMING UP NEXT, A FAST FOOD GODZILLA{,} JOINS THE BURGER WARS IN JAPAN. I'M DAVID BRANCACCIO(.) THIS IS MARKETPLACE. AT THE FOREIGN DESK IN SAN FRANCISCO, I'M GEORGE LEWINSKI. AMERICAN POPULAR CULTURE{.} WHETHER IT'S ROCK AND ROLL[] FASHION[] OR HOLLYWOOD MOVIES, HAS LONG BEEN AN IMPORTANT EXPORT. EVEN THOUGH STATISTICIANS HAVE A HARD TIME MEASURING ITS VALUE. TAKE FAST FOOD. WHEN THE FIRST AMERICAN STYLE BURGER JOINT OPENED IN LONDON'S FASHIONABLE REGENT STREET SOME TWENTY YEARS AGO, IT WAS MOBBED. NOW IT'S ASIA'S TURN(,)

Figure 3 Example of a punctuation generation output. A punctuation generation output is produced for the same part of the TDB98 reference transcription (in single case) by the combined system of a language model and a prosodic feature model (S_LM+CART). (), [] and {} show substitution error, deletion error and insertion error, respectively.

The guardians of the electronic stock market NASDAQ who've been burned by past ethics questions are moving to head off market fraud by toughening the rules for companies that want to be listed on the exchange Marketplace's Philip Boroff reports As part of the proposals Penny stocks will be eliminated from NASDAQ These trade for literally pennies Less than a dollar a share They're the stocks of speculative companies On Wall Street they're the longest of the long shots Some penny stocks grow into established corporations others are shell companies incorporated firms without assets or prospects Some of these are sold by small unsavory brokerage firms That dump them upon gullible investors David Whitcomb is a Rutgers University finance professor and frequent NASDAQ critic that's the real change it's reducing the status of cheap stocks so that at least NASDAQ is not giving them its seal of approval Also these companies will no longer appear in newspapers on NASDAQ's list And Whitcomb says investors may be less prone to buy them If they're not listed in the paper NASDAQ officials say they're not only trying to fight fraud by raising listing standards they're doing a periodic tuneup of their market which they hope will help promote public confidence In New York I'm Philip Boroff for marketplace And that's the top of our news for Thursday November fourteenth Today the Dow Jones industrial average gained thirty eight and three quarter points Details when we do the numbers later on tonight's program life in the fast lane And coming up next a fast food godzilla joins the burger wars in Japan I'm David Brancaccio This is marketplace At the foreign desk in San Francisco I'm George Lewinski American popular culture Whether it's rock and roll fashion or Hollywood movies has long been an important export Even though statisticians have a hard time measuring its value Take fast food When the first American style burger joint opened in London's fashionable Regent street some twenty years ago it was mobbed Now it's Asia's turn

Figure 4 Example of a capitalisation generation output. A capitalisation generation output is produced for the same part of the TDB98 reference transcription by the capitalisation system based on NE recognition and punctuation generation (S_on_NE_P). NE recognition is performed by the rule-based NE recognition system. Punctuation marks are generated by S_LM+CART. Underlined words show the positions of capitalisation generation errors.

The guardians of the electronic stock market <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> who’ve been burned by past ethics questions are moving to head off market fraud, but toughening the rules for companies that want to be listed on the exchange market place is full of <b_enamex TYPE=“PERSON”> Boroff <e_enamex> reports. Is part of the proposals, penny stocks will be eliminated from <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex>. These trade for literally pennies, less than <b_numex TYPE=“MONEY”> a dollar <e_numex> a share. Did the stocks of speculative companies on Wall Street that the longest of the long shots some penny stocks growing to establish corporations, others are shell companies incorporated firms without assets or prospects some of these are sold by small unsavory brokerage firms that dumped them up on gullible investors day that would come as a <b_enamex TYPE=“ORGANIZATION”> Wreckers University <e_enamex> finance professor infrequent <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> credit. That’s the real change, it’s reducing the status of cheap stocks still that at least <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> is not giving them its seal of approval. Also, these companies will no longer appear in newspapers are <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex>’s less than <b_enamex TYPE=“PERSON”> Wiccans <e_enamex> says investors may be less prone to buy them if they’re not listed in the paper <b_enamex TYPE=“ORGANIZATION”> NASDAQ <e_enamex> officials say they’re not only trying to fight fraud by raising listing standards, they’re doing a periodic tuneup of their market which they hope will help promote public confidence in <b_enamex TYPE=“LOCATION”> New York <e_enamex>, I’m <b_enamex TYPE=“PERSON”> Phillip Boroff <e_enamex> for marketplace. And that’s the top of our news for <b_timex TYPE=“DATE”> Thursday, November fourteenth <e_timex>. Today the <b_enamex TYPE=“ORGANIZATION”> Dow Jones <e_enamex> industrial average gained thirty eight and three quarter points details when we do the numbers. <b_enamex TYPE=“PERSON”> Mitterand <e_enamex> tonight’s program life in the fast lane and coming up next the fast food godzilla joined the burger wars in <b_enamex TYPE=“LOCATION”> Japan <e_enamex>, I’m <b_enamex TYPE=“PERSON”> David Brancaccio <e_enamex>. This is market place. The foreign desk in <b_enamex TYPE=“LOCATION”> San Francisco <e_enamex> and <b_enamex TYPE=“PERSON”> George Lewinsky <e_enamex>. American popular culture, whether it’s rock and roll fashion or <b_enamex TYPE=“LOCATION”> Hollywood <e_enamex> movies has long been an important export you know statisticians have a hard time issued its value take a fast food for the first American style burger joint open in <b_enamex TYPE=“LOCATION”> London <e_enamex>’s fashionable regent street some twenty years ago, it was mauled now it’s <b_enamex TYPE=“LOCATION”> Asia <e_enamex>’s turn

Figure 5 Example of a capitalisation generation output for a speech recognition result. NE recognition is performed by the rule-based NE recognition system for the speech recognition results of the HTK system. This speech recognition output contains punctuation marks. Capitalisation generation is performed by S_on_NE_P.

References

[1] 1998 NIST Hub-4 Information Extraction (Named Entity) Broadcast News Benchmark Test Evaluation. Available at ftp://jaguar.ncsl.nist.gov/csr98/h4iene_98_official_scores_990107/index.htm.

[2] Hub-4 IE-NE Evaluation Scoring Program. Available at ftp://jaguar.ncsl.nist.gov/csr98/hub4e_98_eval_disc_doc_981214.tar.Z.

[3] LDC Catalog. Available at http://www.ldc.upenn.edu.

[4] NIST CTM transcription file format for sclite processing. Available at ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/infmts.html#ctm_fmt_name_0.

[5] NIST Hub-4 IE scoring pipeline package version 0.7. Available at ftp://jaguar.ncsl.nist.gov/csr98/official-IE-98_scoring.tar.Z.

[6] The Message Understanding Conference Scoring Software User's Manual. Available at http://online.muc.saic.com/scorer/Manual/manual.html.

[7] *Proceedings of 4th Message Understanding Conference*. Morgan Kaufmann, 1992.

[8] *Proceedings of 5th Message Understanding Conference*. Morgan Kaufmann, 1993.

[9] *The Chicago Manual of Style, 14th Edition*. The University of Chicago Press, 1993.

[10] Named Entity Task Definition. In *Proceedings of the 6th Message Understanding Conference*, pages 317–332, 1995.

[11] *Proceedings of 6th Message Understanding Conference*. Morgan Kaufmann, 1995.

[12] *Proceedings of 7th Message Understanding Conference*. Morgan Kaufmann, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[13] A Universal Transcription Format (UTF) Annotation Specification for Evaluation of Spoken Language Technology Corpora. Available at http://www.nist.gov/speech/tests/bnr/hub4_98/utf-1.0-v2.ps, 1998.

[14] 1999 Information Extraction - Entity Recognition Evaluation. Available at http://www.nist.gov/speech/er_99/er_99.htm, 1999.

[15] J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. MITRE: Description of the ALEMBIC System Used for MUC-6. In *Proceedings of the 6th Message Understanding Conference*, 1995.

[16] S. Abney. Chunks and Dependencies: Bringing Processing Evidence to Bear on Syntax. *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164, 1995.

[17] S. Abney. Partial Parsing via Finite-state Cascades. In *Proceedings of the European Summer School in Logic, Language and Information*, pages 8–15, 1996.

[18] J. Allen. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, 1995.

[19] D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kameyama, and M. Tyson. SRI: Description of the JV-FASTUS System Used for MUC-5. In *Proceedings of the 5th Message Understanding Conference*, pages 221–235, 1993.

[20] D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. FASTUS: A Finite-state Processor for Information Extraction from Real-world Text. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1172–1178, 1993.

[21] D. Appelt and D. Martin. Named Entity Extraction from Speech: Approach and Results Using the TextPro System. In *Proceedings of the DARPA Broadcast News Workshop*, pages 51–54, 1999.

[22] D. Beeferman, A. Berger, and J. Lafferty. Cyberpunc: A Lightweight Punctuation Annotation System for Speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 689–692, 1998.

[23] D. Bikel, S. Miller, and R. Schwartz. Nymble: a High-Performance Learning Name-finder. In *Proceedings of the Applied Natural Language Processing*, pages 194–201, 1997.

[24] W. Black, F. Rinaldi, and D. Mowatt. FACILE: Description of the NE System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[25] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1983.

[26] E. Brill. *A Corpus-Based Approach to Language Learning*. PhD thesis, University of Pennsylvania, 1993.

[27] E. Brill. Some Advances in Rule-Based Part of Speech Tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 722–727, 1994.

[28] E. Brill. Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. In *Proceedings of the Natural Language Processing Using Very Large Corpora*, 1997.

[29] C. Chen. Speech Recognition with Automatic Punctuation. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 447–450, 1999.

[30] H. Chen, Y. Ding, S. Tsai, and G. Bian. Description of the NTU System Used for MET2. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[31] N. Chinchor. MUC-7 Named Entity Task Definition (version 3.5). In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[32] N. Chinchor. Overview of MUC-7/MET-2. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[33] N. Chinchor, P. Robinson, and E. Brown. Hub-4 IE-NE Task Definition Version 4.8. Available at http://www.nist.gov/speech/hub4_98/h4_iene_task_def.4.8.ps, 1998.

[34] K. Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceeding of the 2nd Conference on Applied Natural Language Processing*, pages 136–143, 1988.

[35] P. Clarkson and R. Rosenfeld. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 2207–2710, 1997.

[36] M. Collins. Three Generative, Lexicalised Models for Statistical Parsing. In *Annual Meeting of the Association for Computational Linguistics*, pages 16–23, 1997.

[37] M. Collins. *Head-driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[38] M. Collins and Y. Singer. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[39] A. Conkie, G. Riccardi, and R. Rose. Prosody Recognition from Speech Utterances Using Acoustic and Linguistic Based Models of Prosodic Events. In *Proceedings of the European Conference on Speech Communication and Technology*, 1999.

[40] A. Derouault and B. Merialdo. Language Modelling at the Syntactic Level. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1373–1375, 1984.

[41] E. Dougherty. *Probability and Statistics for the Engineering, Computing and Physical Sciences*. Prentice Hall, 1990.

[42] E. Ejerhed. Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods. In *Proceedings of the 2nd Conference on Applied Natural Language Processing*, pages 219–227, 1988.

[43] M. Fach. A Comparison Between Syntactic and Prosodic Phrasing. In *Proceedings of the European Conference on Speech Communication and Technology*, 1999.

[44] J. Fukumoto, F. Masui, M. Shimohata, and M. Sasaki. Oki Electric Industry: Description of the Oki System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[45] R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of Sheffield: Description of the LaSIE System as used for MUC-6. In *Proceedings of the 6th Message Understanding Conference*, pages 207–220, 1995.

[46] Y. Gotoh and S. Renals. Information Extraction from Broadcast News. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, 358:1295–1310, 2000.

[47] Y. Gotoh and S. Renals. Sentence Boundary Detection in Broadcast Speech Transcripts. In *Proceedings of the International Workshop on Automatic Speech Recognition*, pages 228–235, 2000.

[48] Y. Gotoh, S. Renals, and G. Williams. Named Entity Tagged Language Models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 513–516, 1999.

[49] R. Grishman and B. Sundheim. Design of the MUC-6 Evaluation. In *Proceedings of the 6th Message Understanding Conference*, pages 1–11, 1995.

[50] D. Hakkani-Tur, G. Tur, A. Stolcke, and E. Shriberg. Combining Words and Prosody for Information Extraction from Speech. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 1991–1994, 1999.

[51] J. Hirschberg and C. Nakatani. Acoustic Indicators of Topic Segmentation. In *Proceedings of the International Conference on Spoken Language Processing*, 1998.

[52] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of Sheffield: Description of the LaSIE-II System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[53] C. Huyck. Description of the American University in Cairo's System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[54] S. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[55] J. Kim and P. C. Woodland. A Rule-based Named Entity Recognition System for Speech Input. In *Proceedings of the International Conference on Spoken Language Processing*, pages 521–524, 2000.

[56] J. Kim and P. C. Woodland. Rule Based Named Entity Recognition. Technical Report CUED/F-INFENG/TR.385, Cambridge University Engineering Department, 2000.

[57] J. Kim and P. C. Woodland. The Use of Prosody in a Combined System for Punctuation Generation and Speech Recognition. In *Proceedings of the European Conference on Speech Communication and Technology*, 2001. To appear.

[58] F. Kubala, R. Schwartz, R. Stone, and R. Weischedel. Named Entity Extraction from Speech. In *Proceedings of the Broadcast News Transcription and Understanding Workshop*, pages 287–292, 1998.

[59] C. J. Leggetter and P. C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, 9:171–185, 1995.

[60] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance Measures for Information Extraction. In *Proceedings of the DARPA Broadcast News Workshop*, pages 249–252, 1999.

[61] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[62] W. Mendeltall, D. Wackerly, and R. Scheaffer. *Mathematical Statistics with Applications*. Duxbury Press, 1981.

[63] A. Mikheev. A Knowledge-free Method for Capitalized Word Disambiguation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 159–166, 1999.

[64] A. Mikheev, C. Grover, and M. Moens. Description of the LTG System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[65] A. Mikheev, M. Moens, and C. Grover. Named Entity Recognition without Gazetteers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.

[66] D. Miller, R. Schwartz, R. Weischedel, and R. Stone. Named Entity Extraction from Broadcast News. In *Proceedings of the DARPA Broadcast News Workshop*, pages 37–40, 1999.

[67] S. Miller, M. Crystal, H. Fox, L. Ramshaw, and R. Schwartz. Algorithms that Learn to Extract Information. BBN: Description of the SIFT System as Used for MUC-7. In *Proceedings of*

the 7th Message Understanding Conference, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[68] T. Niesler, E. Whittaker, and P. C. Woodland. Comparison of Part-Of-Speech and Automatically Derived Category-Based Language Models for Speech Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 177–180, 1998.

[69] J. Odell, P. C. Woodland, and T. Hain. The CUHTK-Entropic 10xRT Broadcast News Transcription System. In *Proceedings of the DARPA Broadcast News Workshop*, pages 271–275, 1999.

[70] D. Pallett, J. Fiscus, J. Garofolo, A. Martin, and M. Przybocki. 1998 Broadcast News Benchmark Test Results: English and Non-English Word Error Rate Performance Measures. In *Proceedings of the DARPA Broadcast News Workshop*, pages 5–12, 1999.

[71] D. Palmer, J. Burger, and M. Ostendorf. Information Extraction from Broadcast News Speech Data. In *Proceedings of the DARPA Broadcast News Workshop*, pages 41–46, 1999.

[72] D. Palmer, M. Ostendorf, and J. Burger. Robust Information Extraction from Automatically Generated Speech Transcriptions. *Speech Communication*, 32:95–110, 2000.

[73] M. Przybocki, J. Fiscus, J. Garofolo, and D. Pallett. 1998 Hub-4 Information Extraction Evaluation. In *Proceedings of the DARPA Broadcast News Workshop*, pages 13–18, 1999.

[74] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[75] L. Rabiner and B. Juang. An Introduction to Hidden Markov Model. *IEEE Acoustics, Speech and Signal Processing Magazine*, 3:4–16, 1986.

[76] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[77] S. Rayson, D. Hachamovitch, A. Kwatinetz, and S. Hirsch. Autocorrecting Text Typed into a Word Processing Document. 1998. U.S. patent 5761689. Available at http://www.delphion.com.

[78] S. Renals, Y. Gotoh, R. Gaizauskas, and M. Stevenson. Baseline IE-NE Experiments Using the SPRACH/LASIE System. In *Proceedings of the DARPA Broadcast News Workshop*, pages 47–50, 1999.

[79] H. Shaw. *Punctuate it Right!* Harper-Collins, 1993.

[80] E. Shriberg, R. Bates, A. Stolcke, P. Taylor, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. Ess-Dykema. Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech? *Language and Speech*, 41(3-4):439–487, 1998.

[81] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg. ToBI: A Standard for Labelling English Prosody. In *Proceedings of the International Conference on Spoken Language Processing*, pages 867–870, 1992.

[82] A. Stolcke, E. Shriberg, D. Hakkani-Tur, G. Tur, Z. Rivlin, and K. Sonmez. Combining Words and Speech Prosody for Automatic Topic Segmentation. In *Proceedings of the DARPA Broadcast News Workshop*, pages 61–64, 1999.

[83] B. Sundheim. Overview of Results of the MUC-6 Evaluation. In *Proceedings of the 6th Message Understanding Conference*, pages 13–31, 1995.

[84] P. Taylor and A. Black. Assigning Phrase Breaks from Part-of-Speech Sequences. *Computer Speech and Language*, 12(2):99–117, 1999.

[85] P. Taylor, S. King, S. Isard, and H. Wright. Intonation and Dialog Context as Constraints for Speech Recognition. *Language and Speech*, 41(3-4):489–508, 1998.

[86] R. Weischedel, M. Meteer, R. Schwartz, L. Ramshaw, and J. Palmucci. Coping with Ambiguity and Unknown Words through Probabilistic Models. *Computational Linguistics*, 19(2):359–382, 1993.

[87] M. Wightman. A Stochastic Approach to Named-Entity Extraction. Master's thesis, University of Cambridge, 1998.

[88] P. C. Woodland, T. Hain, S. Johnson, T. Niesler, E. Whittaker, and S. Young. The 1997 HTK Broadcast News Transcription System. In *Proceedings of the Broadcast News Transcription and Understanding Workshop*, 1998.

[89] P. C. Woodland, T. Hain, G. Moore, T. Niesler, D. Povey, A. Tuerk, and E. Whittaker. The 1998 HTK Broadcast News Transcription System: Development and Results. In *Proceedings of the DARPA Broadcast News Workshop*, pages 265–270, 1999.

[90] R. Yangarber and R. Grishman. NYU: Description of the Proteus/PET System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1997. Available at http://www.muc.saic.com/proceedings/muc_7_toc.html.

[91] S. Young. Large Vocabulary Continuous Speech Recognition: A Review. *IEEE Signal Processing Magazine*, 1996.

[92] S. Young and G. Bloothooft. *Corpus-Based Methods in Language and Speech Processing*. Luwer Academic Publishers, 1997.

[93] S. Young, J. Jansen, J. Odell, D. Ollason, and P. C. Woodland. *The HTK book (for HTK version 2.0)*. Cambridge University, 1996.