

# Narrowing the gap between automatic and human word recognition

Cover design: Gies Bouwman & Odette Scharenborg  
Printed and bound by PrintPartners Ipskamp, Nijmegen

ISBN: 90-9019591-2  
© 2005, Odette Scharenborg

# Narrowing the gap between automatic and human word recognition

een wetenschappelijke proeve op het gebied van de Letteren

## Proefschrift

ter verkrijging van de graad van doctor  
aan de Radboud Universiteit Nijmegen  
op gezag van de Rector Magnificus prof. dr. C.W.P.M. Blom,  
volgens besluit van het College van Decanen  
in het openbaar te verdedigen  
op vrijdag 16 september 2005  
des ochtends om 10.30 uur precies

door

**Odette Elizabeth Scharenborg**

geboren op 26 april 1977  
te Groenlo

Promotores: Prof. dr. L. Boves  
Prof. dr. A. Cutler

Co-promotores: Dr. L. ten Bosch  
Dr. J. M. McQueen (Max Planck Instituut voor Psycholinguïstiek)

Manuscriptcommissie: Prof. dr. H. Baayen (Voorzitter)  
Prof. dr. R. K. Moore (University of Sheffield, Groot-Brittannië)  
Dr. D. Jurafsky (Stanford University, Verenigde Staten)

## Een paar woorden van dank

Al vroeg tijdens mijn studie Taal, Spraak & Informatica was het voor mij duidelijk dat ik na de afronding van mijn studie graag wilde promoveren. Zelfs het onderzoeksgebied was voor mij al duidelijk: de spraaktechnologie. Nu is dan mijn proefschrift, iets meer dan 9,5 jaar na mijn eerste dag als student, eindelijk klaar. Natuurlijk heb ik dit proefschrift niet zonder de hulp van velen kunnen doen. Velen hebben mij op verschillende manieren geholpen en deze sectie van mijn proefschrift is dan ook speciaal om hen te bedanken.

Ten eerste is daar Lou Boves, mijn promotor. Toen ik tegen het einde van mijn afstuderen in 2000 aangaf dat ik graag wilde promoveren heeft hij mij het project ‘Psycholinguïstisch plausibele automatische spraakherkenning’ aangeboden. Hoewel ik zelf in eerste instantie twijfelde, geloofde Lou in mij en is dat gedurende de looptijd van het project blijven doen. Ik wil daarom Lou heel hartelijk bedanken voor al het vertrouwen en de hulp gedurende de iets meer dan 4 jaar die het mij ‘gekost’ heeft om dit proefschrift af te ronden. Ten tweede wil ik Louis ten Bosch bedanken voor de 3 jaar die hij mij begeleidt heeft. Ik wens iedere promovendus zo’n enthousiaste en kundige begeleider! Als ik zelf niet te enthousiast was over mijn resultaten of de voortgang, was Louis dat altijd wel. Mijn project bevond zich tussen de twee onderzoeksgebieden van de automatische spraakherkenning en de psycholinguïstiek. Zonder de geweldige begeleiding van James McQueen had ik nooit zoveel kennis opgestoken over de menselijke spraakherkenning. Ik waardeer het zeer dat hij zoveel tijd en energie in mij(n project) heeft gestoken. Tot slot wil ik Anne Cutler, mijn tweede promotor, heel hartelijk bedanken voor de samenwerking. Haar enthousiasme voor mijn project hield mij ook erg enthousiast.

During those four years of my PhD project, I also had the great opportunity to work with some inspiring people from abroad. First of all, I would like to thank Dennis Norris for the interesting collaboration which resulted in two papers which are part of this thesis. And I would like to thank him for hosting me for two months while I was visiting the Medical Research Council – Cognition and Brain Sciences Unit in Cambridge, UK. Secondly, I would like to thank Stephanie Seneff for hosting me for three months while I was visiting the Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA. It was a great and stimulating collaboration which resulted in one paper which is part of this thesis.

Verder wil ik alle mensen waar ik in de loop der jaren mee heb mogen samenwerken bedanken voor hun samenwerking. Met name mijn vroegere en huidige collega’s van de leerstoelgroep Taal & Spraak wil ik bedanken voor alle hulp die ze me boden in de vorm van het lezen van papers, het geven van commentaar op posters en oefenpraatjes, maar

natuurlijk ook voor de gezellige lunches, koffiepauzes (met lekkers!), de borrels, de etentjes, en ga zo maar door. In het bijzonder wil ik Diana Binnenpoorte, Andrea Diersen, Inge de Mönnink en Nelleke Oostdijk bedanken, omdat hun deuren en oren altijd open stonden als ik mijn hart even wilde luchten of gewoon zin had in een gezellig praatje.

Zonder ontspanning geen inspanning en daarom wil ik mijn vrienden bedanken voor alle gezelligheid in de kroeg, op #303, in Lux, of gewoon lekker thuis op de bank. In het bijzonder wil ik Anuska Heutinck en Tom Hendrikx noemen, omdat zij mijn paranimfen wilden zijn! Verder wil ik ook de dames van mijn volleybalteam bij Pegasus heel erg bedanken voor alle gezelligheid en sportiviteit. Maar de meeste dank gaat toch wel uit naar mijn ('schoon'-)familie. Mam, pap en Ellis: bedankt voor jullie steun en dat jullie altijd geïnteresseerd luisteren als ik weer eens vol verhalen over mijn werk zit. Fijn dat jullie altijd in mij hebben geloofd! 'Schoon'familie: bedankt voor jullie interesse en de vele goede gesprekken! En tot slot wil ik Gies heel erg bedanken voor zijn liefde, geduld, steun en hulp, zonder hem had ik niet geweten hoe een automatische spraakherkenner nu eigenlijk werkt.

# Table of contents

Een paar woorden van dank	v
<b>Chapter 1 – Introduction</b>	<b>1</b>
1.1 The recognition of human speech	2
1.1.1 A common goal – Different approaches	2
1.1.2 Narrowing the gap	3
1.2 Research approaches	4
1.2.1 Human word recognition	4
1.2.2 Automatic speech recognition	7
1.3 The issues	10
1.3.1 The input representation	10
1.3.2 Incremental vs. integrated search	11
1.3.3 Word activations vs. path-based scores	12
1.4 The proposed solutions	12
<b>Chapter 2 – Extending Shortlist to an end-to-end model of human speech recognition</b>	<b>15</b>
2.1 Introduction	16
2.2 The joint model	17
2.2.1 Automatic phone recogniser (APR)	17
2.2.2 Shortlist	17
2.3 Material	18
2.3.1 Acoustic data	18
2.3.2 Lexicons	18
2.4 Experiment I: Baseline	18
2.5 Experiment II: Accounting for pronunciation variation	19
2.6 Experiment III: Adjusting the mismatch parameter	19
2.7 General discussion	20
2.8 Conclusion	21
<b>Chapter 3 – How should a speech recogniser work?</b>	<b>23</b>
3.1 Introduction	24
3.1.1 A common goal	25
3.1.2 Human speech recognition	25
3.1.3 Automatic speech recognition	26
3.1.4 Summary	27
3.2 Computational analysis of word recognition	27

3.2.1	Prelexical and lexical levels of processing	27
3.2.2	Cascaded prelexical level	32
3.2.3	Multiple activation and evaluation of words	34
3.2.4	Continuous speech recognition	37
3.2.5	Cues to lexical segmentation	38
3.2.6	No feedback from the lexical level to the prelexical level	40
3.2.7	Summary	42
3.3	SpeM	42
3.3.1	Prelexical and lexical levels of processing	43
3.3.2	Cascaded prelexical level	45
3.3.3	Multiple activation and bottom-up evaluation of words	45
3.3.4	Segmentation of continuous speech	48
3.3.5	Lexical competition and word activation	49
3.3.6	No feedback from the lexical level to the prelexical level	54
3.3.7	Summary	54
3.4	Recognition of words given real speech input	55
3.4.1	Method	57
3.4.2	Results and discussion	57
3.5	Recognition of words in continuous speech	60
3.5.1	Temporarily lexically ambiguous input	60
3.5.2	Lexical competition in spoken word recognition	63
3.5.3	The PWC and the segmentation of continuous speech	66
3.6	General discussion	71
3.6.1	Value of SpeM enterprise for HSR	72
3.6.2	Value of SpeM enterprise for ASR	75
3.6.3	Limitations of the computational analysis	76
3.6.4	Conclusion	77

## Chapter 4 – ‘Early recognition’ of polysyllabic words in continuous speech 79

4.1	Introduction	80
4.2	The recognition system	83
4.2.1	The automatic phone recogniser	84
4.2.2	The search module	85
4.3	The computation of word activation	87
4.4	Material	90
4.5	Early recognition	91
4.5.1	The performance of SpeM as a standard speech recognition system	92
4.5.2	Recognition point analysis	93
4.6	Predictors for reliable on-line early recognition	95



4.6.1	Decision module	96
4.6.2	$\theta$ and $Act_{min}$ as predictors of on-line early recognition	97
4.6.3	The effect of the length of the word	98
4.6.4	Summary	102
4.7	General discussion and conclusion	102

## Chapter 5 – A two-pass approach for handling OOVs in a large vocabulary recognition task 105

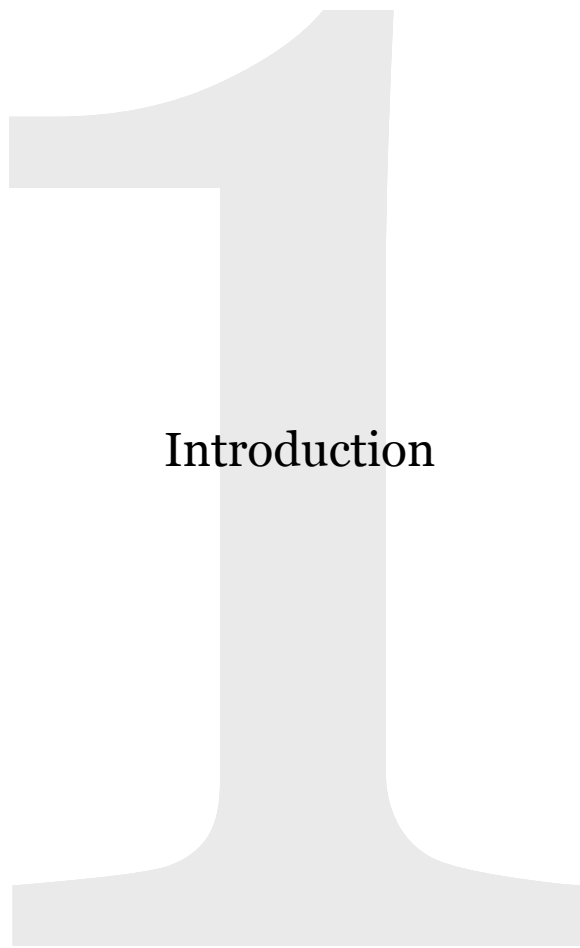
5.1	Introduction	106
5.2	The proposed two-stage recognition system	108
5.2.1	Automatic speech recognition system	109
5.2.2	SpeM	111
5.3	Experimental set-up and materials	113
5.3.1	Experimental set-up	113
5.3.2	Materials	113
5.4	Extracting the subset from the fallback lexicon	114
5.4.1	No utterance-dependent language models	114
5.4.2	Adding utterance-dependent language models	115
5.4.3	Analysis and discussion	116
5.5	Performance of the two-stage recogniser	116
5.5.1	The Ohtsuki method	117
5.5.2	The Tang method	118
5.5.3	Analysis and discussion	119
5.6	Conclusion and future work	119

## Chapter 6 – General discussion and concluding remarks 121

6.1	General discussion	122
6.1.1	Towards an end-to-end model of human speech recognition	122
6.1.2	Incremental vs. integrated search	124
6.1.3	One-stage vs. multi-stage recognition systems	127
6.1.4	Word activation vs. path-based scores	127
6.2	Future work	128
6.3	Concluding remarks	130

Bibliography	133
Summary	143
Samenvatting (Summary in Dutch)	151
Curriculum vitae	159
List of publications	161





# Introduction

*“The central issues in the study of speech recognition by human listeners (HSR) and of automatic speech recognition (ASR) are [...] clearly comparable; nevertheless, the research communities that concern themselves with ASR and HSR are largely distinct.”*

- R. K. Moore & A. Cutler (2001)

*“Given the relatively advanced state of psycholinguistics and speech perception, it seems remarkable that the only working models of lexical access from acoustic waveforms are products of the engineering technology of automatic speech recognition [...]”*

- T. M. Nearey (2001)

## 1.1 The recognition of human speech

In everyday life, speech is all around us, on the radio, television, and in human-human interaction. Communication using speech is easy. We human beings are continually confronted with novel utterances that speakers select from the infinite set of possible utterances in a language, and usually we encounter little to no difficulty in recognising and understanding them. Utterances are made up from a much smaller (but open) set of lexical forms (e.g., word, morphemes). The most important aspect of the process of understanding spoken language by human listeners is the mapping of the information in the speech signal onto word representations in the mental lexicon (e.g., McQueen, 2004), although human listeners also use para- and extra-linguistic information (e.g., Hawkins, 2003). Word recognition is therefore a key component of the speech recognition process. Then, on the basis of the recognised words, an interpretation of the utterance can be constructed.

### 1.1.1 A common goal – Different approaches

There are various research fields that investigate (parts of) the speech recognition process. In this thesis, I focus on two: the fields of human speech recognition (HSR) and of automatic speech recognition (ASR). Although the two research areas are closely related – they both study the speech recognition process, and the central issue of both is word recognition – their aims are different. In HSR research, the goal is to understand how we, as listeners, recognise spoken utterances. This is done by building models of HSR, which can be used for the simulation and explanation of the human speech recognition process. In ASR, the central issue is minimising the number of recognition errors. Much research effort in ASR has therefore been put into the development of systems that generate reliable lexical transcriptions of acoustic speech signals. In parallel with the difference in aims between the two research fields, the research approaches are different as well. To clarify the differences, the research approaches used in HSR and ASR are described in some detail in Section 1.2.

Between the two fields, another difference exists. Although both ASR and HSR claim to investigate the whole recognition process from the acoustic signal to the recognised units, an automatic speech recogniser necessarily is an end-to-end system – it must be able to recognise words from the acoustic signal – while most models of HSR describe only parts of the human speech recognition process. An integral model covering all stages of the human speech recognition process does not yet exist. One part of the recognition process that virtually all models of human speech recognition lack is the part that converts the acoustic signal into some kind of segmental representation. Existing symbolic HSR models cannot recognise real speech, because they do not take the acoustic realisation as their starting point. This makes it hard to evaluate the theoretical assumptions underlying models of HSR in real-life test conditions. Moreover, the HSR models cannot be tested with exactly the same stimulus materials as the human listeners in psycholinguistic studies. Section 1.2 further elaborates on these issues. It describes the approaches used in ASR to recognise speech from the acoustic signal, and it explains the solutions chosen in HSR to deal with the fact that models of HSR cannot recognise real speech.

### **1.1.2 Narrowing the gap**

Despite the gap that separates the two fields, there is a growing interest in possible cross-fertilisation (Moore & Cutler, 2001; ten Bosch, 2001). Specific strands in HSR research hope to deploy ASR approaches to integrate partial modules into a convincing end-to-end model (Nearey, 2001). Nearey (2001) further suggests combining dynamic pattern recognition techniques from ASR with computational models of HSR in order to be able to use “detailed phonetic models [...] as front ends for reasonable models of lexical access”. From the point of view of ASR, there is some hope to improve performance by incorporating essential knowledge about HSR into current ASR systems (Carpenter, 1999; Hermansky, 2001).

The central goal of this thesis is to narrow the gap that has existed for decades between the two research fields of HSR and ASR. Given the central role of word recognition in both ASR and HSR, the focus of this thesis is on word recognition. Taking into account the shortcomings of current computational models as explained above and the suggestions made by researchers in the field of HSR, it is an obvious choice to start the endeavour by trying to build an end-to-end model of human word recognition. Following Nearey’s (2001) suggestion, we do so by using techniques from the field of ASR. The issues that need to be tackled in building such an end-to-end model of HSR using techniques from ASR are described in detail in Section 1.3. Finally, Section 1.4 provides an overview of the remaining chapters of this thesis, in which the solutions we suggest for dealing with these issues are described. But, first, background information on the two research fields is presented.

## 1.2 Research approaches

### 1.2.1 Human word recognition

To investigate the properties underlying the human speech recognition process, HSR experiments with human subjects are usually carried out in a laboratory environment. Subjects are asked to carry out various tasks, such as:

- *Auditory lexical decision*: Spoken words and non-words are presented in random order to a listener, who is asked to identify the presented items as a word or a non-word (Goldinger, 1996).
- *Phonetic categorisation*: Identification of unambiguous and ambiguous speech sounds on a continuum between two phonemes (McQueen, 1996).
- *Sequence monitoring*: Detection of a target sequence (larger than a phoneme, smaller than a word), which may be embedded in a sentence or list of words/nonwords, or in a single word or nonword (Frauenfelder & Kearns, 1996).
- *Gating*: A word is being presented in segments of increasing duration and subjects are asked to identify the word being presented and to give a confidence rating after each segment (Grosjean, 1996).

In these experiments, various measurements are taken, such as reaction time, error rates, identification rates, and phoneme response probabilities. Based on these measurements, theories about specific parts of the human speech recognition system are developed or refined. To put the theories to further test, they are implemented in the form of computational models for the simulation and explanation of HSR. Various models of HSR (e.g., Luce et al., 2000; Marslen-Wilson, 1987; McClelland & Elman, 1986; Norris, 1994) have been developed that are capable of simulating experimental data, obtained through, for instance, word recognition and phoneme perception experiments. In this thesis, the focus is on the process of word recognition. One of the most powerful models for the simulation and explanation of human word recognition is the *Shortlist* model (Norris, 1994).

Most data on human word recognition involve measures of how quickly or accurately words can be identified. A central requirement of any model of human word recognition is, therefore, that it should be able to provide a continuous measure (usually referred to as ‘activation’ or ‘word activation’) associated with the strength of different lexical hypotheses over time. During the human speech recognition process word hypotheses that overlap in time compete with each other. This process is referred to as (lexical) competition. The activation of a word hypothesis at a certain point in time is based on its initial activation and the inhibition caused by other activated words. The word activation score, then, can be compared to the performance of listeners in experiments where they are required to make word-based decisions (such as the above-described auditory lexical decision experiments).

There are two major theories of human speech recognition. The first theory, referred to as ‘episodic’ or sub-symbolic theory, assumes that each lexical unit is associated with a large number of stored acoustic representations (e.g., Goldinger, 1998; Klatt, 1979, 1989). On the other hand, symbolic theories of human speech recognition say that human listeners first map the incoming acoustic signal onto prelexical representations, e.g., in the form of phonemes, after which the prelexical representations are mapped onto the lexical representations (e.g., Gaskell & Marslen-Wilson, 1997; Luce et al., 2000; McClelland & Elman, 1986; Norris, 1994). The speech recognition process in symbolic theories thus consists of two levels: the prelexical level and the lexical level. A central requirement of symbolic computational models is thus a segmental representation of the speech signal. However, as explained before, most HSR models lack a module that converts the speech signal into a segmental representation; instead they use a handcrafted ‘error-free’ linear representation of the input – in the sense that the input always perfectly aligns with the segmental representations of the words in the lexicon. Thus in effect, in most symbolic computational models, the process of creating the prelexical representations is only assumed, and not physically present. Only the output of the prelexical process is available in the form of the handcrafted segmental representation of the speech signal.

This shortcoming could, however, be solved if such an ‘error-free’ representation of the speech signal could be generated automatically. The handcrafted input could then be replaced by the ‘real’ representation of the speech signal. But is it likely that such an ‘error-free’ representation of the speech signal can be (automatically) created? There are reasons to believe that no unique segmental representation of the speech signal exists. One of these reasons is that no absolute truth exists as to what phones a person has produced; therefore, it is not possible to obtain a unique and ‘true’ symbolic transcription of a given speech signal (Cucchiari, 1993). Furthermore, studies in phonetics “suggested that the more detailed a transcription is, the less reliable it tends to be” (Ball & Rahilly, 2002). This statement is backed-up by experiments by, for instance, Shriberg et al. (1984). Shriberg et al. report on a consensus transcription procedure. Two experienced transcribers created a narrow consensus transcription of continuous speech samples of 72 children. These speech samples consisted of approximately five minutes of free conversation. Six weeks after the last tape had been transcribed they created a new narrow consensus transcriptions of 25 utterances for each of eight randomly selected speech samples. Four weeks later, another eight speech samples were randomly selected and transcribed. Comparing the original consensus transcriptions and the retest transcriptions segment by segment yielded an agreement of 68%. However, the percentage agreement went up to 76% when the diacritics were removed from the transcriptions. So, it seems unlikely that the ideal segmental representation of the speech signal can be generated. But how would a symbolic computational model of HSR behave if the best possible (automatically generated) segmental representation were used? The answer (presented in Chapter 2) is as to be expected: it does not perform well.

*The Shortlist model and the theory underlying it*

The research presented in this thesis is based on Shortlist and more specifically the theory underlying the Shortlist model. Shortlist aims at simulating and explaining the lexical processes in word recognition. Like all other HSR models, Shortlist is a partial model – it recognises words given a sequence of phoneme symbols. Chapter 3 details the theory underlying the model, but for the sake of clarity, a summary of that theory is given here:

- *Prelexical and lexical levels:* The incoming acoustic signal is mapped onto prelexical representations, after which the prelexical representations are mapped onto the lexical representations.
- *Autonomous model:* Autonomous models only have a feed-forward flow of information; this is also referred to as ‘bottom-up’ processing. This means that the processing at the phoneme level is totally unaffected by lexical-level processing: Information only flows from the phonemes to the lexical level; no information flows back from the lexical level to the phonemes.
- *Time-shift invariance:* A word can start at any point in time.
- *Competition:* At the lexical level time-overlapping words compete with each other.

In order to implement a theory in the form of a computational model of human word recognition, assumptions have to be made. In Shortlist, these are the following:

- *Input:* The input consists of a single string of discrete phoneme symbols.
- *Output:* The output consists of the activations of all words that were activated by the input.
- *Phoneme mismatch:* When a phoneme is presented in the input that is not in correspondence with a phoneme in an activated word, this word’s activation is decreased.
- *Inhibition:* During lexical competition, word nodes in a neural network are activated in proportion to their match to the input. Words that derive their evidence from the same input phonemes are connected together via inhibitory links. The word with the highest activation will therefore inhibit words with lower activation during competition.
- *Recognition:* The word with the highest activation is recognised.

Word recognition in Shortlist is realised as follows. Each incoming phoneme is matched against the phonemic representations of all words in the internal lexicon. Using an exhaustive lexical search, a candidate list is built containing the words that are roughly consistent with the bottom-up input. This list of candidates is called a shortlist; hence, the name Shortlist. The bottom-up activation of each candidate word is determined by its degree of fit with the input. If a phoneme in a word matches the input, the word activation is increased by 1, for each mismatching phoneme the word activation is reduced by 3 (in the default operation mode of the model). Subsequently, the candidate words are wired into



a neural network. The activated words in the shortlist compete with each other by means of a combined effect of their initial activation and the inhibition by other activated words. This word-word inhibition is proportional to the number of phonemes by which the words overlap. In Figure 1-1, the words connected by arcs compete with each other. Ultimately, the word with the highest activation is recognised.

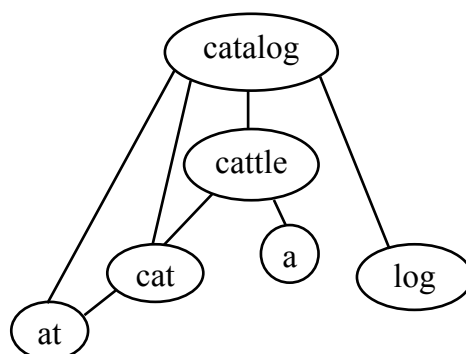


Figure 1-1. The pattern of inhibitory connections between candidates produced by presentation of /kætəlg/ (taken from Norris, 1994).

Shortlist has been successful in the simulation of results obtained in various behavioural studies, such as studies related to the segmentation of a speech stream in words (Norris, 1994; Norris et al., 1997) and lexical competition (McQueen et al., 1994).

### 1.2.2 Automatic speech recognition

The aim of ASR research is to build an algorithm that is able to recognise speech utterances automatically, under a variety of conditions, with the least possible number of recognition errors. Figure 1-2 shows a schematic representation of the steps of the automatic speech recognition process and the information that is needed for it. The speech recognition process is the search for the (hopefully) correct word (sequence) in a search space effectively spanned by the acoustic models, the language models, and the lexicon.

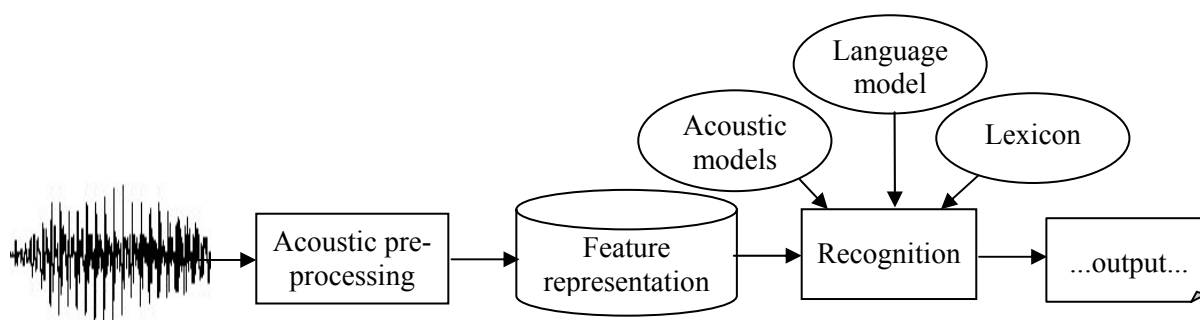


Figure 1-2. A schematic representation of the speech recognition process.

The input of an automatic speech recogniser consists of an acoustic signal. During speech recognition, the speech signal is, first, passed through the acoustic pre-processor where feature vectors are extracted from the speech signal. Subsequently, the features, representing the acoustic signal, are matched with the succession of acoustic models associated with the items, usually words, in the internal lexicon. For each observation, the degree of fit between the observation and each of the models is determined. Ultimately, the word that belongs to the sequence of acoustic models for which the degree of fit is best is hypothesised.

In the match, only those sequences of acoustic models are used that correspond with the words in the lexicon; words not included in the lexicon cannot be recognised. It is important to point out that a standard ASR system is not capable to decide that a stream of feature vectors belongs to a non-word: an ASR system will always come up with a solution in terms of the items that are available in the lexicon<sup>1</sup>. Each word in the lexicon is built from a limited number of units, e.g. phonemes or syllables. The type of units used to describe the words in the lexicon is identical to the type of units represented by the acoustic models. So, if the acoustic models represent phonemes, the units used to describe the words in the lexicon are also phonemes. The language model incorporates the linguistic information that can be learned from the occurrences and co-occurrences of words in a training set.

For each *path*, i.e., a (sequence of) word(s), through the search space, a score is calculated. At the output of the ASR system, the path with the best score is given. In addition, also a word graph can be constructed, which is a compact and efficient representation for storing an *N*-best list. It contains those path hypotheses whose scores are closest to the best scoring path.

For recognition, most ASR systems use an *integrated search*: all information (from the acoustic model set, lexicon, and language model) is used at the same time. The likelihood of a number of hypothesised word sequences (paths) through the complete graph is computed, and then a backtrack is performed to identify the words that were recognised on the basis of the hypothesis with the highest score at the end of the utterance (or the hypothesis with the highest score after a number of recognised words, depending on the length of the influence of the language model). (For more information on automatic speech recognition systems and search, cf., Jelinek (1997)).

ASR systems are usually evaluated in terms of accuracy, the percentage of the input utterances that is recognised correctly, or in terms of word error rate (WER):

---

<sup>1</sup> It is possible to configure an ASR system such that it rejects inputs if the quality of the match with the words in the vocabulary is below a certain minimum. However, this is not the same as detecting that the input speech contains non-words.

$$WER = \frac{\#insertions + \#deletions + \#substitutions}{\#words} \bullet 100\%, \quad (1-1)$$

where  $\#insertions$ ,  $\#deletions$ , and  $\#substitutions$ , are the number of inserted, deleted, and substituted words, respectively, and the  $\#words$  denotes the number of words in the reference transcription.

### *One-stage vs. multi-stage recognition systems*

Automatic speech recognition can be performed in a single step (or stage) or in multiple subsequent stages. In the past, multi-stage ASR systems have been built in which a first stage recogniser converted the acoustic signal into an intermediate representation (of, for instance, phones) after which the second stage recogniser mapped the intermediate representation onto lexical representations (e.g., Klatt, 1977). However, the success of this type of systems was limited. One of the major problems of this approach was that it was very difficult to compute posterior probabilities for the elements in the output of the first stage. In the absence of those probabilities, ‘hard decisions’ about the identity of the phones needed to be made, introducing a lot of errors in the representation, since the first stage appeared far from perfect. Therefore, most ASR systems that were built in the past decades directly map the acoustic signal onto the lexical representations. In this way, the problem with the ‘hard decisions’ is removed, or at least delayed up to a point later in time.

However, lately, a new trend in ASR is to go back to the idea of multi-stage ASR systems. The difference is that this time the intermediate representation of the speech signal is not a deterministic string of phones, but a phone graph, which is a probabilistic representation of the speech signal. The advantage of probabilistic phone graphs over deterministic phone strings is that no hard decisions about the identity of the phones in the phone graph need to be made. It was found that although the correct solution is not always on the first-best path through the phone graph, it often does occur in the top  $N$  best paths available in the phone graph. It can therefore be worthwhile to divide the recognition process (again) into multiple stages. The advantage of such multi-stage recognition systems is that in a second (or subsequent) recognition step more detailed information can be used, for instance by integrating more powerful language models (e.g., morphological, morpho-phonological, morpho-syntactic, and domain knowledge) into the system (see, e.g., Demuyne et al., 2003).

In a second type of multi-stage recognition system, the recogniser is adapted towards the task, the utterance, or language use on the basis of the result of the first recognition step. This result is usually in the form of a word graph or word  $N$ -best lists. After the first stage, a new recognition attempt is carried out, but this time with a tuned recognition system. One way of tuning the second stage of such a multi-stage recognition system is to adapt the lexicon of the second recognition step such that it is more tailored to the utterance to be

recognised (see, e.g., Geutner et al., 1999; Ohtsuki et al., 2004; Tang et al., 2003). The recognition result of the first and second stage are thus of the same type, i.e., words, whereas in the type of multi-stage recognition system described above the segmental representations of the acoustic signal produced by the first and second stages are usually not of the same type, e.g., phones after the first step and words after the second.

Creating intermediate lattice representations of the speech signal is also used in the field of spoken document retrieval. Instead of decoding the acoustic parameter file every time a request is made, a phone or word lattice is created from the acoustics, and a search mechanism searches for the query terms in the stored lattice. This procedure is less time consuming, and it reduces the computational load. For instance, the multi-stage system described by Cardillo et al. (2002) has a first stage that creates a representation of the acoustic signal in the form of a phone lattice which is stored. Subsequently, for each query, a search through this phone lattice on the basis of a lexicon is carried out. Their multi-stage system is thus similar to the multi-stage system described at the start of this section.

### *The automatic phone recogniser as an acoustic front-end*

ASR systems do not adhere to a theory of HSR like computational models of human speech recognition do. Thus, a standard ASR system cannot be used as a computational end-to-end model of human speech recognition. But an ASR system is able to make a segmental representation of the speech signal, making it a seemingly logical step to use an adapted ASR system as the missing front-end that converts the acoustic signal into a symbolic representation in current HSR models. To that end, an ASR system is built that does not recognise words, but recognises phones. Such an ASR system is called an automatic phone recogniser (APR). The lexicon of the APR only contains the phones of the language, and its ‘language model’ now incorporates the occurrence and co-occurrence frequencies of phones. An APR functions the same as a standard ASR system: at the input, the acoustic signal is presented, and at the output a segmental representation of the speech signal is produced in the form of the first-best hypothesis and a graph. But, in the case of an APR, the first-best hypothesis and the graph consist of sequences of phones instead of words.

## 1.3 The issues

Several issues need to be resolved in order to build an end-to-end model of HSR using techniques from ASR. In this section, the main issues are explained in detail.

### **1.3.1 The input representation**

As explained in Section 1.2, in Shortlist a handcrafted segmental representation of the speech signal is presented at the input. Thus, similar to other symbolic computational models of HSR, in Shortlist the process of creating the prelexical representations is only

assumed, and only the *output* of the prelexical level is available in the form of the handcrafted segmental representation of the acoustic signal.

When building our end-to-end model of human word recognition, we, of course, do have to build the prelexical level. This prelexical level, thus, should create the segmental representation of the speech signal, preferably ‘error-free’, i.e., the input should perfectly align with the segmental representations of the words in the lexicon.

However, speakers usually do not adhere to the canonical pronunciations of words when talking. Especially in spontaneous speech, speech sounds may be reduced, deleted, inserted, and substituted compared to the ‘canonical’ pronunciation. As already indicated in Section 1.2.1, it is thus unlikely that a segmental representation of the speech signal can be created that perfectly aligns with the segmental representations of the words in the lexicon, especially since no top-down information about the phonemic representations of the words can be used (Shortlist is an autonomous model).

Based on the finding from ASR that the correct path often features in the top  $N$  best paths in a (phone) graph, it is a logical step in our search for an end-to-end model of HSR to investigate whether replacing the one-dimensional input representation used by current HSR models by a representation of the speech signal in the form of a probabilistic phone graph will bring us closer to our goal of a working end-to-end model of HSR. This approach is described in Chapters 2 and 3.

### 1.3.2 Incremental vs. integrated search

Most mainstream ASR systems use some kind of integrated search algorithm: humans, on the other hand – at least according to obvious behaviour and mainstream psycholinguistic theory on human speech perception – seem to compute an on-line activation measure for words as the speech comes in (and presumably make a decision as soon as the activation of a word is high enough; see also Chapter 3). In order to model the human speech recognition process, computational models of HSR should thus be able to provide word activation scores over time, as the input comes in. So, it should be possible for words to be recognised before their acoustic realisation is complete. In order for an end-to-end model based on ASR techniques to be able to recognise words while the speech comes in, the traditional integrated search cannot be used. A different type of search, one that hypothesises words while the input comes in, e.g., an incremental search, needs to be implemented.

The capability of recognising words on the basis of their initial part helps humans in detecting and processing self-corrections, broken words, repeats, etc. (Stolcke et al., 1999). This makes it worthwhile to investigate whether an ASR system using an incremental search would be able to perform ‘early recognition’, i.e., recognising a word before the end of its acoustic realisation is complete. This is done in Chapter 4.

If a word is not present in an ASR system's lexicon, it cannot be recognised; this will result in an error. This problem of out-of-vocabulary (OOV) words is well-known in ASR. One possible solution would be to include as many words as possible in the recogniser's lexicon. Very large lexicons do not necessarily pose a problem for ASR systems, but the combination with a weak language model usually results in poor performance. If only a weak language model is available, one might want to adopt a strategy in which a subset of words is selected from the large vocabulary to create a smaller lexicon of most likely words. The ability of the incremental search to recognise parts of words can be used for the extraction of such subsets of words from a large vocabulary. This approach is addressed in Chapter 5.

### 1.3.3 Word activations vs. path-based scores

As explained in Section 1.2, human listeners compute *word* activation scores, while ASR systems calculate *path*-based scores. When building an end-to-end model of human word recognition using techniques from the field of ASR, a way needs to be found to relate the path-based ASR scores to the word activation scores in HSR. If a word lies on the first-best path, this does give an estimate of the activation of the word but not its precise value. Related to this issue, during the human speech recognition process words that overlap in time compete with each other. However, in ASR, no active inhibition of words is possible, since only path scores are being calculated. These issues are addressed in Chapters 3 and 4.

The word activation scores as used in HSR and the path-based ASR scores can also be regarded as confidence scores: the higher the word activation score, the more likely that the word is indeed present in the input. This might make such an activation score a useful predictor of early recognition. This is put to the test in Chapter 4.

## 1.4 The proposed solutions

My first attempt in creating an end-to-end computational model of human word recognition on the basis of an APR and Shortlist is described in Chapter 2. The end-to-end model was tested with the best possible (automatically generated) one-dimensional segmental representation that was available.

Chapter 3 describes the second attempt to build an end-to-end model of HSR. This end-to-end computational model of human word recognition (based on the theory of Shortlist) is called SpeM. In SpeM, the one-dimensional segmental representation is replaced by a probabilistic phone graph, and it is completely built using techniques from ASR. In addition to being able to use real speech instead of an idealised form of input, this model also resembles the speech recognition process in human listeners because of the two-step recognition procedure: it has both a prelexical level, at which a segmental representation of the speech signal is created, and a lexical level, at which the segmental representation is mapped onto lexical representations.

The problem that the traditional integrated search in ASR is not able to hypothesise words before their acoustic offsets is solved in SpeM by using an incremental search that gives a ranked list of the most likely words at each point in time while the input comes in, and thus hypothesises words before their acoustic offsets.

In Chapter 3, I propose a method to convert the path-based scores that are used in ASR search methods, and thus also in SpeM, into word-based activation scores. The details of the underlying mathematics are presented in Chapter 4. I further explain how these word-based activation scores can be used for the simulation of the lexical competition process.

The solutions chosen for the issues described in Section 1.3 make it possible to use the end-to-end model of human word recognition presented in Chapter 3 as an unconventional ASR system: besides doing normal speech recognition, it has the capability of doing recognition tasks a human listener can easily perform, but standard ASR systems cannot. In Chapter 4, I investigate how SpeM's incremental search can be used for the task of early recognition, i.e., recognising a word before its acoustic offset. I looked for predictors that can be used to determine during the speech recognition process whether a word is correctly recognised before its acoustic offset. One of the investigated predictors was the word activation score calculated by SpeM.

In Chapter 5, SpeM's ability of recognising word-initial cohorts is used for dealing with the problem of OOV words. A multi-stage recognition system is presented in which a large number of OOVs (in the form of city names) exist at the first stage. On the basis of the recognition results of the first stage recogniser, SpeM is used to select a subset of city names from a larger lexicon containing city names. In the subsequent recognition run, the list of words created by SpeM is used as part of the lexicon. The second stage recogniser is thus tuned to the task. Presumably, the correct city name is no longer an OOV, thus removing the problem of OOVs.

In summary, the research presented in Chapters 2 and 3 shows the benefit that can be obtained by using techniques from the field of ASR for building models of HSR. The experiments described in Chapters 4 and 5 show the benefit for ASR of a recognition procedure that makes use of key aspects of the human speech recognition process.

This thesis ends with a chapter in which the findings of the research presented in Chapters 2 through 5 are discussed and put into perspective. Also, the main conclusions are drawn and suggestions for further research are presented.







## Extending Shortlist to an end-to-end model of human speech recognition

Reformatted from:

O. Scharenborg, L. ten Bosch, L. Boves, and D. Norris (2003). “Bridging automatic speech recognition and psycholinguistics: Extending Shortlist to an end-to-end model of human speech recognition,” *Journal of the Acoustical Society of America*, 114 (6), 3032-3035.

*This letter evaluates potential benefits of combining human speech recognition (HSR) and automatic speech recognition by building a joint model of an automatic phone recogniser (APR) and a computational model of HSR, viz. Shortlist (Norris, 1994).*

*Experiments based on ‘real-life’ speech highlight critical limitations posed by some of the simplifying assumptions made in models of human speech recognition. These limitations could be overcome by avoiding hard phone decisions at the output side of the APR, and by using a match between the input and the internal lexicon that flexibly copes with deviations from canonical phonemic representations.*

## 2.1 Introduction

In this letter, we address speech recognition by making a bridge between two disciplines that have little overlap with respect to theoretical framework and experimental paradigms. One discipline is automatic speech recognition (ASR), which studies the automatic transformation of a speech signal into a sequence of discrete ‘recognition tokens’ (commonly words). The main goal in ASR research is to minimise the number of recognition errors on a certain test set under specific testing conditions. The second discipline is the area of human speech recognition (HSR). In HSR, the conversion from an acoustic signal to (a string of) words is studied with a focus on understanding the psychological processes underlying human word recognition, e.g. the word perception process per se.

In HSR experiments, the usual stimuli are carefully spoken utterances recorded in noiseless environments. On the basis of theories of HSR, several computational models have been developed to simulate data from experiments on human speech perception. These models compute word activations as the input unfolds over time, where activation can be related to the speed and accuracy with which human listeners can recognise words. However, the existing computational models of HSR model only parts of the human speech recognition process. Typically, one of the missing parts is a module that converts the acoustic speech signal into a representation that forms an appropriate input for the models, which almost invariably assume some kind of symbolic representation of the speech signal.

Most experimental studies of HSR are based on read speech; however, in the last few years, the focus is shifting towards (more) spontaneous speech. Much more than read speech, spontaneous speech is affected by articulatory processes such as assimilation and reduction. Since listeners are sensitive to this type of subtle sub-phonemic information (e.g. Gow, 2002; see Cutler, 1998, for an overview), and to durational differences in the input (Davis et al., 2002), HSR models are now challenged to address the question of how the speech signal is mapped onto lexical representations in more detail. This is an area where established techniques from ASR could be useful in informing future research. Nearey (2001) suggests combining dynamic pattern recognition techniques from ASR with HSR models in order to be able to use “detailed phonetic models [...] as front ends for

reasonable models of lexical access”. Nearey doubts that existing HSR models “will work as advertized when attached to real phonetic transduction systems”.

The present letter presents the results of experiments that put Nearey’s conjecture to the test by attempting to make a bridge between the two research areas by studying a combined ASR-HSR model (henceforth referred to as ‘joint model’) that can be regarded as an end-to-end model of human speech recognition. The input for the computational model of HSR is provided by an automatic phone recogniser (APR). This HSR model is tested with input consisting of extemporaneous, ‘real-life’ speech.

## 2.2 The joint model

The proposed joint model is a first step in the development of an end-to-end model of HSR. From the available computational models for human word recognition, we have chosen Shortlist (Norris, 1994) to use in the joint model, because it has been successfully applied to a wide range of data from studies of HSR.

The joint model works as follows. The APR decodes a speech signal into a sequence of phone symbols; Shortlist takes this sequence as input and generates a sorted word list. These processes are discussed in more detail below.

### 2.2.1 Automatic phone recogniser (APR)

For the APR, we trained 36 context-independent (hidden Markov) phone models, one silence model, one model for hesitations such as ‘uh’, and one noise model (Scharenborg et al., 2002a). The APR decoding is based on a phone loop with optional silence preceding and following each phone, and is guided by a phone bigram. The APR output is a purely phonemic representation of the acoustic signal – without word boundaries.

### 2.2.2 Shortlist

In its present implementation, Shortlist itself is a two-stage model. In the first stage, the input (i.e., a sequence of phone symbols) is processed from left to right and an exhaustive search of the internal lexicon yields a shortlist of word candidates (max. 30<sup>1</sup> per phone position) that roughly match the phonemic input processed so far. The ‘activation’ of the words in the shortlist is determined by the ‘degree of fit’ between the phones in the input and the string of phones specified in Shortlist’s internal lexicon. For each phone in the input that matches the lexicon representation of a word, the word’s activation is increased with 1; otherwise the activation is reduced by the mismatch parameter (default value is 3). In the second stage – the competition stage – the candidates in the shortlist enter into a

---

<sup>1</sup> The number 30 is arbitrarily chosen; the exact value does not have a large effect on the performance of the model (Norris, 1994).

network where time-overlapping candidates compete with each other. The output consists of (a sequence of) the most activated word(s).

## 2.3 Material

### 2.3.1 Acoustic data

For training the APR, data from a Dutch telephone corpus (the Dutch Directory Assistance Corpus, DDAC) were used (Sturm et al., 2000). DDAC contains telephone calls to the Dutch 118 Directory Assistance service. Most utterances consist of either one Dutch city name or ‘ik weet het niet’ (‘I don’t know’) pronounced in isolation. Others may also contain disfluencies and longer connected speech fragments. From this corpus, an independent test set (DDAC-test) of 10,510 utterances comprising 11,523 words was selected.

### 2.3.2 Lexicons

The baseline lexicon of Shortlist consists of 2,392 city names and ‘ik weet het niet’ (‘I don’t know’). For each word in the lexicon, one unique ‘canonical’ phonemic representation was available.

The psycholinguistic theory underlying Shortlist makes no claim about the manner in which humans cope with pronunciation variation. Specifically, there is nothing in the theory that promotes the exclusive use of citation forms in the mental lexicon. Therefore, in order to deal with pronunciation variation, we created a second lexicon (‘PronVar’) with on average 2.6 pronunciation variants per word (Scharenborg et al., 2002a).

## 2.4 Experiment I: Baseline

We investigated the performance of the joint model in a baseline experiment using the baseline lexicon. The input for Shortlist consists of the speech utterances of DDAC-test transcribed by the APR. The parameter settings of Shortlist are identical to those used in Norris (1994). The ‘performance’ of the joint model was tested in terms of the ASR benchmarking method of recognition errors, rather than on the psycholinguistic benchmark of similarity to human performance. Thus, the performance measure in this study is word accuracy: the percentage of utterances for which the reference words (in DDAC-test) receive the highest activation value at the output of Shortlist.

With an accuracy of 23.5%, the performance of the joint model in this baseline experiment appears to be quite poor. Since the performance of Shortlist on canonical phone representations is close to 100%, this result shows that recognising real-life speech is more difficult than recognising ‘perfect’ phonemic transcriptions. An error analysis reveals that the model has great difficulty in dealing with reduced forms: the APR output mostly comprises fewer (and sometimes also different) phones than the canonical representation stored in Shortlist’s lexicon.

Two follow-up experiments were carried out. The aim of the experiments was to study the possible improvement of the joint model’s baseline performance using two strategies: using a lexicon that accounts for pronunciation variation (Experiment II), and adjusting the value of the mismatch parameter in Shortlist (Experiment III).

## 2.5 Experiment II: Accounting for pronunciation variation

The second experiment is identical to Experiment I, except that the PronVar lexicon (including pronunciation variations) was used. Using PronVar, Shortlist’s performance as a speech recogniser – reported in terms of word accuracy – increases substantially with 16.2% absolute to 39.7%. An error analysis reveals that there are few cases where the correct word is in the shortlist, but where a competitor receives a higher final activation. This finding suggests that, in the case of non-canonical input, the selection of correct lexical candidates into the shortlist is problematic. This problem is addressed in Experiment III.

## 2.6 Experiment III: Adjusting the mismatch parameter

Listeners are highly sensitive to any mismatch between input phones and the phonological representations of words; a mismatch of a single phonological feature can eliminate all signs that a word has been activated (e.g. McQueen et al., 1999). Because of these findings, Shortlist weights mismatching information much more heavily than matching information. However, a high value of the mismatch parameter could actually impair recognition of real-life speech considerably, as even quite small deviations from the expected lexical representation might make a word unrecognisable.

*Table 2-1.* Effect of  $M=3.0$  and  $M=0.0$  measured in terms of the accuracy and the percentage of utterances for which the correct word was present in the shortlist (% In shortlist). Two lexicons are used, viz. baseline and PronVar.

<b>Mismatch</b>	<b>Baseline lexicon</b>		<b>PronVar lexicon</b>	
	<i>Accuracy (%)</i>	<i>In shortlist (%)</i>	<i>Accuracy (%)</i>	<i>In shortlist (%)</i>
3.0	23.5	24.3	39.7	42.3
0.0	32.5	59.5	54.1	76.5

In experiment III, we investigated the effect of ‘cancelling’ the mismatch penalty ( $M$ ) by setting  $M=0.0$  in a test with both lexicons (for a complete account of the experiment, see Scharenborg et al., 2002b). Table 2-1 shows the results in terms of the percentage of utterances for which the correct word is present in the shortlist (‘In Shortlist’). In addition, we report the word accuracy of the joint model on the word recognition task.

The first row of Table 2-1 shows the results of Experiment II for reference. As can be seen in Table 2-1, using  $M=0.0$  increases the model's performance with both lexicons compared to the default value  $M=3.0$ .

## 2.7 General discussion

The aim of the research described in this letter is to build and evaluate an end-to-end computational model of HSR – based on a joint model of an APR and Shortlist – that takes acoustic recordings of ‘real-life’ speech as input. Real-life speech is characterised by pronunciation variation, which leads to non-canonical phonemic representations. In order to study the effects of non-canonical input to Shortlist, we carried out three experiments. Experiment I was the baseline experiment. In short, Experiment II showed that including pronunciation variants in the internal lexicon of Shortlist improves the ability of the joint model to deal with real-life input. Experiment III showed that the combination of a mismatch parameter value of 0.0 and the use of the lexicon containing pronunciation variants is best able to deal with the reduced phonemic forms encountered in real-life speech. This combination yields a recognition accuracy of 54.1%, which is more than twice the baseline performance.

The experiments show that a straightforward combination of an APR and Shortlist does not yield an end-to-end model of HSR that can deal satisfactorily with real-life input, despite the fact that the APR and Shortlist each perform well in their own domains. Apparently, one cannot take for granted that a combination of the best models of two sides yields the best overall end-to-end model. Perhaps this is not too surprising, since neither system was designed with the intention of being interfaced with the other. Nevertheless, these experiments illustrate the consequences of some of the simplifying assumptions made in Shortlist and other HSR models, and show the extent to which these assumptions need to be revised to produce genuine end-to-end models that will be able to deal with the pronunciation variation present in spontaneous speech.

One shortcoming of the joint model is that it makes ‘hard’ decisions both at the level of input phones, and in the goodness-of-fit metric used in the search process. Shortlist requires a single string of phone symbols as input. This implies that the APR is forced to make ‘hard’ decisions about the segmental representation of the speech signal based only on the acoustic information. Also for HSR (e.g. Gaskell et al., 1998; McQueen et al., 1999), data from experiments indicate that human listeners do not make hard decisions prior to lexical selection. This problem with Shortlist has been addressed in the Merge model (Norris et al., 2000), which is derived from Shortlist. However, the present implementation of Merge can handle only very small lexicons. One can eliminate hard decisions in the input by representing the speech signal as a segment-based lattice containing multiple segment-string hypotheses. The subsequent word search or activation algorithm should make the final decision which phones were present by re-ranking the activated words or taking the first best.

The second level of ‘hard’ decisions involves the word search process in Shortlist. This search matches input phone strings to the phone strings stored in the lexicon in a way that it is intolerant of deviations from the canonical form of words. This is exactly the problem highlighted by Nearey (2001) and is certainly an area where more flexible pattern-matching techniques (such as dynamic programming as commonly used in ASR) could play an important role in refining computational HSR models. Of course, the resulting refined model should still be able to simulate actual data of HSR experiments.

An important question to be borne in mind when assessing the results of our experiments is whether our conclusion would have been radically different had we been able to drive Shortlist with the output of a human ‘phone recogniser’ rather than the APR or with the output of an APR optimised on the task. Cucchiarini et al. (2001) showed that automatically generated transcriptions of read speech are very similar to manual phonetic transcriptions created by expert phoneticians. Such transcriptions are to a large extent also non-canonical. Thus, transcriptions created by human expert transcribers would cause similar problems for HSR models. In Scharenborg et al. (2002b), it is shown that optimising the APR settings in order to improve the balance between generating an input phone sequence that is close to the signal and at the same time meets the input criteria of Shortlist does not improve the performance of the joint model. So, while our experiments may not provide a precise quantitative measure of the extent of the problems faced by Shortlist, the problems are real nonetheless.

Finally, we would like to raise an additional point<sup>2</sup>. A human being is able to identify a non-lexical token as a nonword. However, the joint model is not able to classify any input as a nonword, since it simply activates the nearest known word. Identification of a nonword could be made possible by using an activation threshold: when no lexical token exceeds the threshold, the system identifies a nonword. This is one topic for further research.

## 2.8 Conclusion

This letter describes a coupling of an automatic phone recogniser and a computational model of human word recognition, viz. Shortlist. The coupling helped to identify aspects of the two components of the joint model that need to be improved in order to build a comprehensive end-to-end computational model of HSR that is able to deal with real-life speech. One of the future research directions is extending the representation of the speech signal from a single linear input phone string to a probabilistic phone graph. This allows, in a natural way, the postponement of a hard decision to a point later in the word search process, which we believe is desirable. A second possibility of improvement lies in changing the current word search in Shortlist into a search algorithm based on dynamic

---

<sup>2</sup> This issue was raised by one of the anonymous reviewers of an earlier version of this letter.

programming techniques. By doing so, deviations from the canonical representations can be dealt with in a natural way.

### Acknowledgements

Earlier results and parts of the research presented in this article are published in the proceedings of the ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology, Estes Park, Colorado, USA, 2002, and in the proceedings of the 7<sup>th</sup> International Conference on Spoken Language Processing, Denver, Colorado, USA, 2002.

The authors would like to thank Anne Cutler, James McQueen, and Roel Smits for fruitful discussions about this research and their comments on earlier versions of this letter. Furthermore, the authors would like to thank the four anonymous reviewers for raising additional interesting issues and giving their useful comments on an earlier version of this letter.





## How should a speech recogniser work?

Reformatted from:

O. Scharenborg, D. Norris, L. ten Bosch, and J. M. McQueen. “How should a speech recognizer work?,” *Accepted for publication in Cognitive Science*.

*Although researchers studying human speech recognition and automatic speech recognition share a common interest in how information processing systems (human or machine) recognise spoken language, there is little communication between the two disciplines. We suggest that this lack of communication follows largely from the fact that research in these related fields has focussed on the mechanics of how speech can be recognised. In Marr's (1982) terms, emphasis has been on the algorithmic and implementational levels rather than on the computational level. In the present paper, we provide a computational-level analysis of the task of speech recognition which reveals the close parallels between research concerned with human and automatic speech recognition. We illustrate this relationship by presenting a new computational model of human spoken word recognition, built using techniques from the field of automatic speech recognition that, in contrast to current existing models of human speech recognition, recognises words from real speech input.*

### 3.1 Introduction

Researchers in the fields of both human speech recognition (HSR) and automatic speech recognition (ASR) are interested in understanding how it is that human speech can be recognised. It might seem, therefore, that this common goal would foster close links between the disciplines. However, while researchers in each area generally acknowledge that they might be able to learn from research in the other area, in practice, communication is minimal. One barrier to communication might be that the research is often seen as being about *how* humans, or *how* machines, recognise speech. In one sense, the answers to these questions must necessarily be different because of the radical differences in the hardware involved (brains vs. computers). However, questions posed at a higher level of analysis may well have the same answers in both disciplines. In his book "Vision" (1982), Marr argues that complex information processing systems can be described at three different levels: the computational, the algorithmic, and the implementational. Computational-level descriptions focus on specifying both *what* functions a particular information processing system must compute, and *why* those computations are required to achieve the goals of the system. In contrast, the algorithmic and implementational levels address the question of *how* computations are performed. The algorithmic level specifies the algorithms and representations involved in the computations, while the implementational level is concerned with how representations and algorithms can be realised physically. From an information processing perspective, Marr suggests that the computational level is the most important. Although speech recognition in humans and machines is implemented in very different ways, at the computational level humans and machines must compute the same functions, as both must perform the same task. Marr himself suggests that a failure to distinguish between *what* and *how* questions has hampered communication between disciplines such as artificial intelligence and linguistics. Exactly the same problem seems to prevent communication between HSR and ASR.

Here, we attempt to construct a computational analysis of the task of recognising human speech. In presenting this analysis, we relate the computational level to the different algorithms used in ASR and HSR. Although ASR uses vocabulary like dynamic programming and pre-processing, whereas HSR is described in terms of lexical competition and auditory perception, we show that most of these terms have direct homologues in the other domain.

As a concrete illustration of the parallels between HSR and ASR we present a new model of HSR constructed using techniques from the field of automatic speech recognition. This new model, called SpeM, can be considered to be an implementation of the Shortlist model (Norris, 1994) with one important difference from Shortlist: SpeM can recognise real speech.

### **3.1.1 A common goal**

In HSR research, the goal is to understand how we, as listeners, recognise spoken utterances. We are continually confronted with novel utterances that speakers select from the infinity of possible utterances in a language. These utterances are made up from a much more limited set of lexical forms (words or perhaps morphemes). The only way a listener can understand the message that is conveyed by any given utterance is thus to map the information in the acoustic speech signal onto representations of words in their mental lexicon, and then, on the basis of stored knowledge, to construct an interpretation of that utterance. Word recognition is therefore a key component of all HSR models.

Word recognition is also a major focus of research in the field of ASR. Although speech-driven systems may have many higher-level components (e.g., for semantic interpretation), these components, just as for human listeners, require input from sufficiently accurate and efficient word recognition. Much research effort in ASR has therefore been put into the development of systems that generate reliable lexical transcriptions of acoustic speech signals.

Given the centrality of word recognition both in human speech comprehension and in ASR systems, we will limit the present discussion to a computational analysis of the word recognition process itself. An account of word recognition at Marr's computational level of description will apply equally well to computer speech systems as to human listeners. Whether the speech recogniser is human or machine, it still has the same computational problem to solve. The principal question we will try to answer, therefore, is this: What computations have to be performed in order to recognise spoken words?

### **3.1.2 Human speech recognition**

Explanatory theories in HSR have generally focussed on quite specific issues such as acoustic variability (e.g., Elman & McClelland, 1986; Stevens, 2002), the lexical segmentation problem (e.g., Norris, McQueen, Cutler & Butterfield, 1997), and the

temporal constraints on the word recognition process (e.g., Marslen-Wilson, 1987; Marslen-Wilson & Welsh, 1978). Many of the more influential psychological models have been implemented as computational models (e.g., Shortlist, Norris, 1994; TRACE, McClelland & Elman, 1986; and the Neighborhood Activation Model, Luce & Pisoni, 1998), and each of these models has had success in simulating important empirical data. However, none of these models attempts to supply a complete account of how the acoustic signal can be mapped onto words in the listener's mental lexicon. Each deals only with particular components of the speech recognition system, and many parts of the models remain unspecified. This is often for the very good reason that there is no constraining data, or no a priori reason to prefer one implementation to another. That is, these are primarily piece-meal approaches; there are no grand unified theories accounting for all aspects of human spoken word recognition. As a consequence, no matter how well a model may be able to simulate the available psychological data, it is difficult to assess whether the assumptions embodied in the model are actually consistent with an effective complete recognition system. For example, in Shortlist, the simplifying assumption is made that the word recognition process receives a sequence of discrete phonemes as input (Norris, 1994). Could such an assumption really hold in a fully functioning recogniser? More importantly, if this simplifying assumption were abandoned, would it have implications for the way other components of the model work? In the context of a restricted model it is difficult to ascertain whether many of the assumptions in these models are plausible. It is therefore necessary to step back from detailed explanations of particular psycholinguistic data sets, and ask, following Marr (1982), how well HSR models address the computational problems that must be solved for successful word recognition.

### **3.1.3 Automatic speech recognition**

In ASR research, it is impossible to avoid confronting all aspects of word recognition simultaneously, such as speaker accents, speaking style, speaking rate, and background noise. The success of a recogniser is generally measured in terms of its accuracy in identifying words from acoustic input. Mainstream ASR approaches are usually implementations of a specific computational paradigm (see below for details), unencumbered by any considerations of psychological plausibility. An ASR system must be able to recognise speech tolerably well under favourable conditions, but nothing in the behaviour of such a system needs to map onto any observable human behaviour, such as reaction times in a listening experiment. Similarly, the representations and processes in ASR systems need not be psychologically plausible; all that matters is that they work. Consequently, any practical ASR system is unlikely to be a candidate for a psychological theory.

### 3.1.4 Summary

In a sense, therefore, we have two complementary models of speech recognition. HSR models explain at least some human behaviour, but often leave a lot to the imagination when it comes to a detailed specification of how the recognition system would actually perform the task of recognising spoken words with the acoustic signal as starting point. In contrast, ASR models can recognise speech, but offer little in the way of explaining human behaviour. HSR and ASR are however not complementary in the set-theoretic sense of being non-overlapping. For either type of model to be a success, it has to be consistent with a computational level description of the problem of recognising speech. In the following section, therefore, we present an analysis of the word recognition process at the computational level, and discuss how both HSR and ASR systems have dealt with different aspects of the word recognition problem. We thus try to bridge the gap that exists between HSR and ASR (Moore & Cutler, 2001) by linking them at the computational level.

## 3.2 Computational analysis of word recognition

### 3.2.1 Prelexical and lexical levels of processing

Two acoustic realisations of the same word, or even the same sound, are never identical, even when both are spoken by the same person. These differences are due to factors such as speaker-dependent characteristics (e.g., vocal tract length, gender, age, speaking style, and emotional state), phonological context (e.g., sounds appearing at different places within a syllable or word are pronounced differently), coarticulation processes, and prosody. Furthermore, speakers usually do not adhere to the canonical pronunciation of words when talking; speech sounds may be reduced, deleted, inserted, and substituted. The resulting pronunciations of those words are often referred to as pronunciation variants. The speech recogniser must be able to accommodate this variability. Humans and computers are thus faced with the task of mapping a highly variable acoustic signal onto discrete lexical representations (such as words). We will refer to this as the ‘invariance problem’ (see, e.g., Perkell & Klatt, 1986). What kind of algorithms and representations could perform the computations required to solve this problem?

One possible solution to the invariance problem is to assume that each lexical unit is associated with a large number of stored acoustic representations, and that these representations cover the normal variability observed in the signal (e.g., Goldinger, 1998; Klatt, 1979, 1989). In the HSR literature, theories that rely on storing representations of each encounter with a word are often called ‘episodic’ theories. Episodic theories of lexical organisation have been successful in explaining experimental data showing that human listeners are able to remember details of specific tokens of words that they have heard, and that such episodic memories for words influence subsequent speech processing (see, e.g., Goldinger, 1998). However, the most obvious limitations of episodic models, especially if they refer to entire words, follow from their inefficiency compared to models using sub-

lexical representations. Learning to recognise a word reliably will require exposure to a large number of acoustic realisations of that particular word. That is, a model that simply stores multiple episodes of words has to learn each word independently. Nothing the model learns about recognising one word will make it any better at recognising previously unencountered words<sup>1</sup>.

A similar issue of generalisation occurs across speakers. It is rather unclear how an episodic word recognition system could robustly recognise speech produced by a new speaker with unusual speech characteristics (e.g., a speaker of an unfamiliar dialect, or a speaker with a speech impediment) without learning new representations for each new word that that speaker utters. Even if the new (unknown) speaker differs from known speakers in a completely systematic and predictable manner, for example by consistently pronouncing one particular phoneme in an unusual way, this systematicity cannot easily be exploited to help recognise words spoken by the new speaker. In order to take account of the systematicity in pronunciation an episodic model would first of all have to be able to analyse both the input and the episodic lexical representations in terms of their sublexical components, and then to modify the episodic representations of all words accordingly. These modified representations would then no longer correspond to any previously encountered episode. However, human listeners can rapidly adapt to a new speaker after exposure to only a few words. Norris et al. (2003) have shown that listeners can quickly learn that a speaker produces a particular phoneme in an unusual manner; moreover, McQueen et al. (in preparation) have shown that this knowledge generalises to the processing of new words not yet heard from that speaker. Such learning seems to require a more abstract level of representation of speech sounds, at a prelexical level of processing. Adjustments made in response to idiosyncratic speech at this level of processing would allow generalisation to novel words. Models with fully episodic lexical representations, however, lack phonologically abstract prelexical representations.

Although there is no doubt that listeners can retain very detailed memories of the acoustic-phonetic properties of individual word tokens, this episodic information cannot support the robust generalisation to new words and speakers shown by human listeners. In contrast, HSR models that rely primarily on abstract representations (such as phonemes or features) are able to generalise to new words and speakers. A drawback to this type of theory,

---

<sup>1</sup> It is well known that episodic models can form abstractions (e.g., Hintzman, 1986). This type of generalisation, however, applies to new tokens of categories that have previously been presented to the model (e.g., new tokens of previously presented words), and not to novel categories (e.g., previously unencountered words). We are therefore not arguing that episodic models are completely unable to generalise. Nevertheless, they are unable to take advantage of what they have learned about the set of words in their previous experience in recognising a novel word.

however, is that they have difficulty explaining how details of specific tokens of words heard and remembered by human listeners can influence subsequent speech processing.

Furthermore, the use of abstract phonological representations at a prelexical level of processing, that is, one that mediates between low-level auditory processing and higher-level lexical processing, helps to address the invariance problem. Prelexical representations such as features, phonemes, or syllables would provide a means of modelling the acoustic-phonetic information in the speech signal in terms of a limited number of sub-word units, and thus offer the possibility for a more efficient coding of the variability in the signal than whole-word episodic models. For example, information about the variability associated with the stop consonant [t] could be associated with a single phonemic representation of that consonant (or perhaps representations of a small number of allophones), rather than with the lexical representations of all words containing [t].

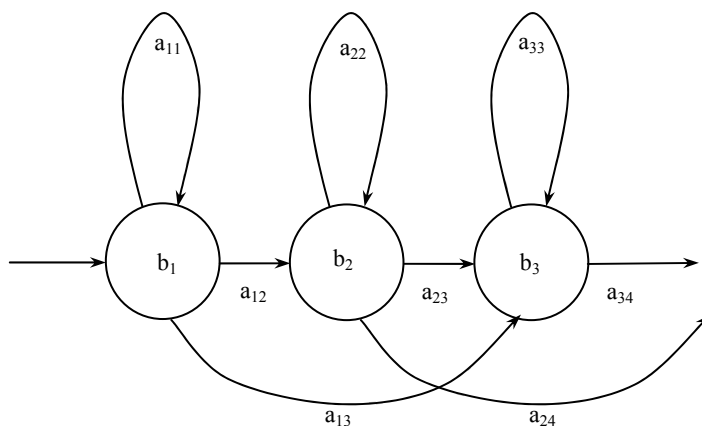
Because of the listener's ability to generalise over new words and speakers, most HSR word recognition models therefore assume that there is some kind of prelexical level. The exact form of the representations at the prelexical level is still the topic of extensive research and debate (see McQueen, 2004, for a review) – in fact, this is arguably the most important question in current HSR research. In the absence of a clear answer to this question, different models make different assumptions about the form that prelexical representations take, for example: phonemes in Shortlist (Norris, 1994); acoustic-phonetic features and phonemes in TRACE (McClelland & Elman, 1986); features in the Distributed Cohort Model (DCM; Gaskell & Marslen-Wilson, 1997); and context-sensitive allophones in PARSYN (Luce et al., 2000).

ASR solutions to the invariance problem in large part parallel those proposed in HSR. Some ASR models have close parallels to episodic models of HSR. In such models, each word is associated with a (large) number of acoustic templates, and it is assumed that these templates cover the variability observed in the signal. Speaker verification by spoken signatures is often based on the processing of a limited number of acoustic word templates (Furui, 1996). For each individual speaker, a few speech samples corresponding to specific words (e.g., spoken passwords, or spoken signatures) are stored, and every time a new speaker is encountered, new speech samples for each word of that new speaker are recorded and stored. However, this kind of approach is not practical when the recognition system is intended to be used by many people or for large vocabularies: Adding new speech samples for each new speaker is often not feasible.

An alternative ASR approach to the invariance problem is to build sub-word statistical models that encode the expected variation in the signal. These sub-word models could in

principal represent several types of speech segments (e.g., phones<sup>2</sup>, syllables, diphones, or triphones). In the lexicon used by the ASR system, each word has one or more representations (i.e., the canonical representation plus possibly pronunciation variants) coded in terms of those sub-word units. Most mainstream mid- and large vocabulary ASR systems are based on statistical phone models (see, e.g., Juang & Furui, 2000; Lesser et al., 1975; Rabiner & Juang, 1993).

In developing such ASR systems, there are two obligatory steps. First, in the front-end, a mapping is made from the raw speech signal to so-called *features* (i.e., numerical representations of speech information). The most important function of these features is to provide a relatively relevant, robust, and compact description of the speech signal. Ideally, the features would preserve all information that is relevant for the automatic recognition of speech, while eliminating irrelevant components of the signal such as those due to background noise. These features describe spectral characteristics such as the component frequencies found in the acoustic input and their energy levels. Second, in the acoustic modelling stage, an acoustic model is created for each recognition unit (e.g., each phone). Such an acoustic model usually consists of a sequence of hidden Markov model (HMM) states (or artificial neurons in the case of an artificial neural network). For an introduction on HMMs, the reader is referred to Rabiner & Juang (1993).



*Figure 3-1.* A graphical representation of a Hidden Markov Model consisting of three states.

---

<sup>2</sup> A phone is the smallest identifiable unit found in a stream of speech that can be transcribed with an IPA symbol. A phoneme is the smallest contrastive phonological unit in the sound system of a language.



Figure 3-1 shows a graphical representation of an HMM consisting of three states (indicated by the circles in Figure 3-1). Each state describes a specific segment of speech using the features that were computed during the feature extraction process. These feature vectors are clustered together and the probability of any given cluster is then described in terms of probability density functions (pdfs; indicated as  $b$  in Figure 3-1). For example, the acoustic model for a particular phone might encode the expected spectral variability that occurs when that recognition unit is spoken in the context of different neighbouring recognition units or when people with different regional accents produce that specific recognition unit. The pdfs are estimated over all acoustic tokens of the recognition unit in the training material. Once trained, the acoustic model can be used to derive an estimate of the probability that a particular stretch of signal was generated from the occurrence of a particular recognition unit ( $P(W|X)$ , where  $P$  denotes the probability,  $W$  is the recognition unit, and  $X$  is the acoustic model of the recognition unit). The variability in duration found in the speech signal is modelled by a set of transition probabilities (indicated by  $a$  in Figure 3-1), namely:

- self-loop (Figure 3-1:  $a_{i,i}$ ): remain in the current state;
- next (Figure 3-1:  $a_{i,i+1}$ ): jump to the next state;
- skip (Figure 3-1:  $a_{i,i+2}$ ): skip one state.

A very common procedure for training acoustic models maximises the likelihood that a given acoustic signal has been generated by a given acoustic model, more precisely, it maximises  $P(X|S)$  (in which  $P$  denotes the probability,  $S$  is the speech model, and  $X$  is the acoustic signal). The procedure for training acoustic models is such that sequences of acoustic models corresponding to sequences of speech segments are trained simultaneously instead of one acoustic model at a time. The trained acoustic models can then be used for recognition. During word recognition, the incoming speech signal is matched against the acoustic representations of the words in the lexicon.

ASR systems with sub-word models have the same advantage as HSR models with prelexical representations: New words can be learned simply by learning the appropriate sequence of sub-word models, and such knowledge will automatically generalise to new tokens of the word. In fact, once a sufficient range of subword models has been trained, new words can be recognised simply by providing a phonemic transcription of those words. No prior exposure to the new words is required. This is exactly how commercial systems such as IBM's Via Voice and ScanSoft's Dragon Dictate work. The recogniser only requires exposure to a representative sample of speech in order to achieve accurate recognition of a large vocabulary.

This comparison of HSR and ASR approaches to the invariance problem already shows that there are strong similarities at the computational level between the two domains.

Because both HSR and ASR researchers are attempting to solve the same computational problem, it should come as no surprise that they have developed similar solutions.

Although we argued above that for successful speech recognition, a prelexical level is needed in order to effectively solve the invariance problem, it is critical to note that the search algorithms in mainstream ASR approaches are generally indifferent to the level of representation or size of the models involved. In the search process, the distinction between prelexical and lexical levels is almost absent. The search for the sequence of words that best matches the signal is usually performed by searching for the best path through a *lattice*. The left hand panel of Figure 3-2 shows an example of a word-based lattice. During the search, the lattice is built dynamically. At the lowest level, the nodes in the lattice are the individual HMM-states (see also Figure 3-1). The connections (or the allowed transitions) between the nodes are fully specified by the combination of HMM model topologies (i.e., the number of states present in the HMM of each sub-word unit), the structure of the word in the lexicon in terms of the sub-word units, and, if applicable, a language model that specifies the syntax, that is, the allowed (possibly probabilistic) ordering of the words in the output of the speech recogniser. This means that in this lattice, the information on the level of probabilistic acoustic detail up to the level of probabilistic linguistic information about syntax is integrated in a single structure, which is used to decode the speech signal in terms of words. A lattice has one begin node (denoted ‘B’ in Figure 3-2) and one end node (denoted ‘E’ in Figure 3-2). There are multiple *paths* from ‘B’ to ‘E’ following the direction of the arrows, and, given an utterance as input, each path corresponds to a possible lexical parse of the input.

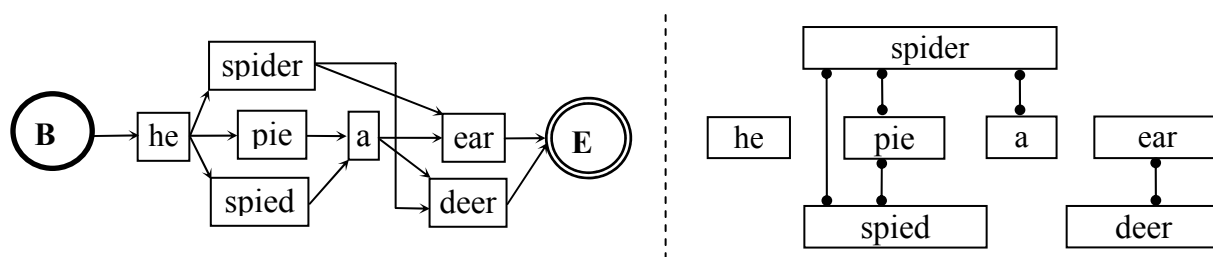


Figure 3-2. The left panel shows an example of a word lattice as used in automatic speech recognition; the right panel shows the competition process that occurs in human speech recognition.

### 3.2.2 Cascaded prelexical level

An ideal speech recogniser would be able to recognise spoken words in close to real time. For the human listener, this is necessary for efficient communication. It indeed appears to be the case that there is very little lag between when a word is spoken and when it is recognised: On the basis of results from a number of different listening tasks, Marslen-

Wilson (1987) estimated this lag to be only 200 ms (i.e., about one syllable at an average speaking rate).

To achieve this rapid recognition, HSR models generally assume that there is continuous, or cascaded, flow of information between the prelexical and lexical levels. That is, rather than sending discrete chunks of information after each prelexical unit is identified, the prelexical level continuously outputs the results of all partial analyses. If these two levels operated serially, with categorical decisions being taken at the prelexical level before lexical access was initiated, this would introduce delays in processing time: The lexical level would have to wait for decisions about each prelexical unit in the input (e.g., about each phoneme or each syllable) before word recognition could be achieved. Cascaded processing helps to avoid this delay. Moreover, as McQueen et al. (2003) have argued, cascaded processing has another benefit with respect to the timing of word recognition: It allows contextual information (i.e., the semantic or syntactic constraints imposed by the preceding words in the utterance) to be used immediately in the process of lexical selection.

Extensive experimental HSR data support cascaded processing. A growing body of HSR experiments has shown that lexical processing is modulated by fine-grained acoustic-phonetic information (e.g., Andruski et al., 1994; Davis et al., 2002; Gow, 2002; Marslen-Wilson & Warren, 1994; McQueen et al., 1999; Salverda et al., 2003; Spinelli et al., 2003; Tabossi et al., 2000; see McQueen et al., 2003, for review). Other HSR research has shown that lexical processing is continuous and incremental (i.e., it changes as the input unfolds over time, e.g., Allopenna et al., 1998; Zwitserlood, 1989). Again, such findings suggest that the prelexical level is not a discrete processing stage.

Although fast and immediate recognition is vital to successful human communication, real-time recognition is not always important in ASR applications. For example, in systems designed for the orthographic transcription of large collections of speech, recognition can satisfactorily be performed off-line (Makhoul et al., 2000). However, although they are not able to match human performance, some ASR systems are able to perform speech recognition in close to real time. For example, commercial systems designed to recognise dictated speech (e.g., Via Voice and Dragon Dictate) can often produce results shortly after words have been uttered. However, the solution that is hypothesised after a certain word has been presented to the system may change based on additional information that arrives after this word, and this adjustment process may delay the output of the eventual hypothesised word sequence. This effect is often due to the influence of long-span language models, for instance, tri-gram language models, which affect the interpretation of bottom-up evidence of individual words from the acoustic signal. A very similar phenomenon is observed in human listeners. When listening to natural speech (as opposed to the read speech used in the studies reviewed by Marslen-Wilson, 1987) listeners may not recognise a word until several following words have been heard (Bard et al., 1988).

In most ASR systems, there is a form of cascaded processing. This can be found in the graded, continuous matching that occurs between the signal and the acoustic models. As explained above, in mainstream ASR approaches, the word search is based on a search of optimal paths through a decoding lattice. This lattice is constructed on the basis of both prelexical and lexical representations (see Section 3.2.1), which means that decisions about the final recognition result are based on the combination of all information in the decoding lattice, rather than just the prelexical level alone. The matching scores that are the result of the matching function all contribute to the *a posteriori* probability of a path in the decoding lattice.

Cascaded processing in ASR is more clearly visible in ASR systems using a two-step approach (e.g., Demuynck et al., 2003). The first step involves the decoding of the incoming acoustic signal into a (probabilistic) lattice with prelexical units, and the second step involves a lexical search (and sometimes even the search for semantic information) from this intermediate lattice representation. The lexical search in the second step does not need to wait for the lattice to be final, but can start while the lattice is still under construction. Again, therefore, because of the computational constraint of needing to recognise speech rapidly, both HSR models and ASR systems have converged on cascaded processing algorithms.

### **3.2.3 Multiple activation and evaluation of words**

At the heart of the problem of speech recognition is the task of matching the representation of the speech signal to words in the lexicon. This task can be considered to have three subcomponents. First, the input must be compared with representations of the words in the lexicon. Second, some assessment must be made of the degree to which the input matches those representations. Finally, a choice must be made as to which word matches the input most closely. In HSR, these three subcomponents can sometimes correspond to separable parts of a model. In ASR, these subcomponents are generally combined in a single search process, as described earlier (see Section 3.2.1).

The matching process necessarily depends on the form of the prelexical representations. In an HSR model like Shortlist, where the prelexical representations are simply a sequence of phones, the process of comparing the input to lexical representations is trivial: It is performed by a serial search through the lexicon. However, in common with almost all psychological models, it is assumed that human listeners can perform this search in parallel. The degree of match between the input and individual representations is calculated very simply in the Shortlist model: words score +1 for each matching phoneme, and -3 for each mismatching phoneme. A set of words that best matches the input (the Shortlist) is then entered into an interactive activation network. The word nodes in the network are activated in proportion to their match to the input as determined by the match/mismatch score, and words that derive their evidence from the same input phonemes are connected together via inhibitory links. The word with the highest activation (i.e., the greatest

perceptual support) will therefore inhibit words with lower activation during competition, and the best matching word will be recognised. As we will see later, however, the real value of the competition process is in the recognition of continuous speech.

The matching process in TRACE is more complicated than in Shortlist because the TRACE network has featural nodes as well as phoneme nodes, and there is competition (inhibition) between phoneme nodes as well as word nodes. The most important difference between TRACE and Shortlist is probably that, in contrast to Shortlist, word activation is not decreased by the presence of mismatching phonemes. Both models assume that access to lexical entries occurs in parallel, that word nodes in an interactive activation network are activated in proportion to their degree of match to the input, and that selection of the best matching word is achieved by competition between activated words.

In the Cohort model (Marslen-Wilson, 1987, 1990; Marslen-Wilson & Welsh, 1978), the input signal activates all words that begin in the same way as the input (e.g., start with the same phoneme as the input). These words form what is called the *word-initial cohort*. Whenever the mismatch between the input and the target word becomes too large, the candidate word drops out of the cohort. A word is recognised via a decision process where the activation values of the words that remain in the cohort are compared. Recognition takes place when the difference in activation between the best candidate word and its runner-up exceeds a certain criterion. In all of these HSR models (and others, such as the Neighborhood Activation Model, Luce & Pisoni, 1998, the DCM, Gaskell & Marslen-Wilson, 1997, and PARSYN, Luce et al., 2000) there is therefore some kind of competition mechanism which allows for the relative evaluation of multiple candidate words.

A large number of psycholinguistic experiments, using a wide variety of different paradigms, have amassed considerable evidence that multiple candidate words are indeed ‘activated’ simultaneously during human speech comprehension (e.g., Allopenna et al., 1998; Gow & Gordon, 1995; Tabossi et al., 1995; Zwitserlood, 1989). There is also extensive evidence that there is some form of relative evaluation of those alternatives (e.g., Cluff & Luce, 1990; McQueen et al., 1994; Norris et al., 1995; Vitevitch & Luce, 1998, 1999; Vroomen & de Gelder, 1995). The data on both multiple activation and relative evaluation of candidate words are reviewed in McQueen et al. (2003) and McQueen (2004).

The competition mechanism in HSR models helps them solve what we will refer to as the ‘lexical embedding’ problem. Because natural language vocabularies are large (many languages have on the order of 100,000 words), but are constructed from a limited set of phonemes (most languages have inventories of between 10 and 50 phonemes; Maddieson, 1984), and since words have a limited word length, it is necessarily the case that there is considerable phonological overlap among words. Any given word is likely to begin in the same way as several other words (Luce, 1986), and to end in the same way as other words. In addition, longer words are likely to have shorter words embedded within them

(McQueen et al., 1995). This means that when the recogniser is presented with any fragment of a spoken word, that fragment is likely to be compatible with many lexical alternatives.

Parallel evaluation of lexical hypotheses is thus the main solution to the lexical embedding problem in HSR. Note that the particular choice of algorithms in HSR is strongly influenced by the belief that the brain is a parallel processing device, which is therefore capable of comparing many different lexical hypotheses simultaneously. For our present purposes, it is worth drawing attention to an important terminological consequence of contrast between parallel activation theories and serial search models. In the psychological literature, activation models are often thought to stand in contrast to search processes. However, in the ASR literature, the entire speech recognition process is seen largely as a search problem: How should an ASR system search through the entire set of lexical hypotheses to discover which best matches the input? The search might be performed serially, or in parallel, depending on the choice of algorithms and hardware. Technically then, even parallel activation models in psychology are really search models.

In ASR, the search module searches for the word that maximises the likelihood of the word given the speech signal:  $P(W|X)$ ; in which  $P$  is the probability,  $W$  is the word, and  $X$  is the acoustic signal. The search is often implemented by a dynamic programming (DP) technique (e.g., Rabiner & Juang, 1993). Two often-used types of DP are A\* search (e.g., Paul, 1992), in which the best hypothesis is searched for in a time-asynchronous depth-first way, and Viterbi decoding, in which the search strategy is time-synchronous and breadth-first (e.g., Rabiner & Juang, 1993; for a text-book account, see Chapter 5 of Jelinek, 1997). Both of these algorithms are simply efficient methods of finding the best path through a lattice, that is, the sequence of words that best matches the input. During the processing of incoming speech, pruning techniques remove the most implausible paths, in order to keep the number of paths through the search space manageable. As a result, only the most plausible words are considered in the search.

During the recognition of isolated words, multiple paths (corresponding to candidate words in HSR) are considered simultaneously, and each candidate word (or to be more precise: path) is assigned a *score* that indicates the match between the word and the input. Internally, the paths – or candidate words – and their corresponding scores are ranked on the basis of the path score. The path (which in the case of isolated word recognition will only contain one word) with the best score wins. The score each path obtains is determined on the basis of Bayes' Rule and is related to 1) the probability that the acoustic signal is produced given the word ( $P(X|W)$ ), and 2) the prior probability of the word (usually based on its frequency of occurrence;  $P(W)$ ).

Thus, while in ASR systems the lexical access and lexical selection stages are combined into one search module, pruning mechanisms do have the effect of limiting the search. This has parallels in the Shortlist model, where only a small number of words are considered as

candidates at each point in time. Similarly, the search process in ASR, and the ordering of surviving paths in the lattice on the basis of the accumulated path scores, are akin to the relative evaluation processes seen in HSR models. There is one important difference between the ASR approach and psychological models, however. In HSR models such as Shortlist and TRACE, the competition process involves individual lexical candidates, whereas most ASR systems base their search on comparing scores of complete paths. Nevertheless, this qualification aside, there are again strong similarities between ASR and HSR.

### 3.2.4 Continuous speech recognition

So far, our computational analysis of spoken word recognition has focussed on the task of identifying isolated words. However, the real task facing a listener, or an automatic speech recogniser, is to identify words in utterances. Natural utterances are continuous, with no gaps or reliably marked boundaries indicating where one word might end and another begin. That is, to a first approximation, *speech is comparable to handwritten text without spaces* (thus, with no gaps between words or letters). Words in a spoken utterance may therefore in principle start and end at any time in the acoustic signal. In itself, the absence of clear boundaries might not create any additional computational problems beyond that involved in isolated word recognition. If all words in the language were highly distinctive, and could be identified at or before their final phoneme, then words could be identified in sequence, with one word starting where the next ended (as in the Cohort model). However, in natural languages this is not the case. The lexical embedding problem (described in the previous section) is particularly acute given continuous speech as input. Consider the input *ship inquiry* ([ʃɪpɪŋkwɪəri]). Within this phone sequence several words can be found starting and ending at different moments in time, for example, *ship*, *shipping*, *pink*, *ink*, *inquiry*, and *choir*. In addition, there are a multitude of partially matching words, such as *shin*, *sip*, *shipyard*, *pin*, *quite*, *quiet*, and *fiery*. How do we determine the best way of parsing this continuous input into a sequence of words? Once again, this can be considered to be a search problem.

The algorithms for continuous speech recognition used in HSR and ASR are usually rather different. However, in both cases, the algorithms are direct extensions of the procedures used for isolated word recognition. As noted earlier, a critical difference between ASR and HSR models is that search in ASR is based on hypotheses at the utterance level (i.e., paths through the lattice for all the words in an input sentence), while the evaluation process in HSR is at the word level (e.g., competition between individual lexical hypotheses). This difference is illustrated in Figure 3-2. The left hand panel shows a graphical representation of a set of words in the form of a lattice with possible paths through the utterance as used in ASR, while the right hand panel shows a graphical representation of the same set of activated words and the inhibitory connections between those words, as in HSR models such as TRACE and Shortlist.

In HSR models, the best parse of the input is generated by the lexical competition process. Because lexical candidates that overlap in the input inhibit each other, the most strongly activated sequence of words will be one in which the words do not overlap with each other. Also, because words with more bottom-up support have more activation, the competition process will tend to favour words that completely account for all phonemes in the input over any sequences that leave some phonemes unaccounted for. Through the competition process, the activation value of a given candidate word comes to reflect not only the goodness of fit of that word with the input with which it is aligned, but also its goodness of fit in all lexical parses of the utterance that it is involved in. The competition process thus results in the optimal segmentation of the input. Lexical competition is therefore a valuable algorithm in HSR, both for the lexical embedding problem, and for the segmentation problem.

The ASR algorithm for the recognition of sequences of words is also an extension of the algorithm for the recognition of isolated words. In the case of isolated word recognition, each path corresponds to one word; in the case of continuous speech recognition, each path corresponds to a word or a sequence of words. The underlying search algorithm is identical. In the case of continuous speech recognition, the score on the basis of which the paths are ranked and the best path is determined is based on three factors (instead of two in the case of isolated words): 1) the probability that the acoustic signal is produced given the word sequence ( $P(X|Path)$ , in which *Path* denotes a word or word sequence); 2) the prior probability of each word (based on its frequency of occurrence), and 3) possibly other higher level sources of information with respect to the recognition unit and its context (like N-gram scores or grammars).

It is worth noting that in the original account of the Shortlist model (Norris, 1994), it was suggested that the ‘lexical competition’ process could equally well be performed by a dynamic programming algorithm instead of an interactive activation model, and Figure 3-2 indeed clearly shows the striking resemblance between the search in ASR and the competition process in HSR models. This reinforces the point that competition and search are simply alternative algorithms that perform the same computational function.

### **3.2.5 Cues to lexical segmentation**

Work in HSR has suggested that there is more to the segmentation of continuous speech than lexical competition. Although they are not fully reliable, there are cues to likely word boundaries in the speech stream (e.g., cues provided by rhythmic structure, Cutler & Norris, 1988; phonotactic constraints, McQueen, 1998; acoustic and allophonic cues, Church, 1987; and silent pauses, Norris et al., 1997), and listeners appear to use these boundary cues in segmentation. The question therefore arises how this boundary information can be used to modulate the competition-based segmentation process in HSR models. Norris et al. (1997) argue that human listeners use a lexical viability constraint called the Possible Word Constraint (PWC). As implemented in Shortlist, the PWC



operates as follows: Each candidate word is evaluated with respect to any available cues to likely word boundary (i.e., boundaries marked by rhythmic, phonotactic, allophonic, and acoustic signals). If the stretch of speech between the edge of a candidate word and the location of a likely word boundary is itself not a possible word, then that candidate word is penalised (its activation is halved). A stretch of speech is not a possible word if it does not contain a vowel. Cross-linguistic comparisons have suggested that this simple phonological constraint on what constitutes a possible word in lexical segmentation may be language universal (see, e.g., Cutler et al., 2002).

Norris et al. (1997) suggested that an important benefit of the PWC was that it would help solve the problem of recognising speech containing unknown or “out-of-vocabulary” words. There are many reasons why a portion of an utterance may not match any lexical entry (due, e.g., to a mispronunciation, to masking of part of the signal by noise, to use of an unknown pronunciation variant, or of course to use of a genuine out-of-vocabulary word). Competition-based recognisers will tend to parse such inputs in terms of the words that are in the lexicon. Consider the utterance “They met a fourth time”, but spoken by a speaker of a London dialect of English, who produces the last sound of the word *fourth* as [f]: *fourth* will thus be said as *fourf*. As Norris et al. argue, a competition-based model such as Shortlist, if *fourf* is not stored as a word form in the lexicon, will tend to recognise such a sequence as *They metaphor f time*. This is clearly inadequate. What is required is a mechanism which will generate plausible candidates for new word forms (such as *fourf*) and rule out impossible new word forms (such as *f*). The PWC achieves this: candidates such as *metaphor* and *four* will be penalised because there is a vowelless sequence (the single *f*) between the end of those words and the boundary marked at the onset of the strong syllable *time*. The sequence *fourf* will thus be available as a potential new word form, perhaps for later inclusion in the lexicon (see Norris et al. for more details and simulation results).

Most ASR systems do not have a mechanism that looks for cues in the speech signal to help the segmentation process. However, a few attempts have been made to use prosodic cues to help the segmentation process. In analogy with the finding of Norris et al. (1997) that human listeners use silent pauses to segment speech, Hirose et al. (2001) have built an ASR system that uses silent pauses to place boundaries between morae in Japanese speech. The number of attempts to use prosodic (or other types of) cues in the segmentation process in ASR is small, however, and the results are usually poor. A mechanism like the PWC has not to our knowledge yet been added to ASR models.

In HSR, the PWC is supported by experimental data and is motivated as an algorithm to solve the out-of-vocabulary problem. The out-of-vocabulary problem is usually not a big problem in ASR systems that have been developed for a specific (small) task, such as digit recognition. However, when the task involves a more natural conversation between a human and an ASR system, such as an automatic directory assistance system where the

caller can ask for any business or private telephone listing, the number of out-of-vocabulary words increases dramatically, reducing the recognition accuracy of the ASR system. A number of ASR systems therefore have a mechanism to detect out-of-vocabulary words (e.g., Hypothesis Driven Lexical Adaptation, HDLA; Waibel et al., 2000): If a sequence cannot be associated with a lexical entry with any high degree of certainty, it will be labelled as out-of-vocabulary, and usually not processed further (there are exceptions, for instance, the second pass in HDLA does process the entries labelled as out-of-vocabulary). However, the detection method is prone to errors. Furthermore, few of those systems can automatically ‘learn’ these out-of-vocabulary words. Adult human listeners, however, can learn new words from limited exposure, and these words appear to be rapidly incorporated into the listener’s lexicon (Gaskell & Dumay, 2003). As we have just argued, the PWC can assist in this learning process through helping to specify which novel sequences in the input are potential new words. There is therefore a fundamental difference between human and machine speech recognition. HSR must be an inherently dynamic process (i.e., must be able to change over the course of the listener’s lifetime), while ASR systems are usually built for a specific purpose, and thus, after an initial training and development phase, are basically fixed systems. That is, HSR algorithms must be flexible, in order for the listener to be able to deal with the changing speech input. ASR systems may need to become more flexible if they are to be able to achieve large vocabulary speaker-independent recognition. The PWC could offer a mechanism in ASR for more dynamic handling of out-of-vocabulary words.

### **3.2.6 No feedback from the lexical level to the prelexical level**

During word recognition in a model with prelexical and lexical levels of processing, information must flow bottom up from the acoustic signal to the prelexical level, and from there to the lexical level. A question that is still unanswered, however, and one that is rather controversial within the field of HSR, is whether information also flows from the lexical level back to the prelexical level. Norris et al. (2000) argued that there was no psycholinguistic data which required the postulation of a lexical feedback mechanism, and argued that some data (that of Pitt & McQueen, 1998) challenge HSR models such as TRACE, which have feedback. Furthermore, Norris et al. pointed out that this kind of feedback as a word is heard could not help recognition of that word, and could in fact harm recognition of sub-lexical units within that word such as phonemes.

It is important to note that this debate concerns the architecture of the speech recognition system and not whether lexical and prelexical processes both contribute to word recognition. All researchers agree that both lexical and prelexical information contribute to the final interpretation of the speech signal. The question about feedback, therefore, is that, if there are separate processes responsible for lexical and prelexical processing, does information from a lexical processor feed back to influence the operation of the prelexical

processor? This question is still hotly debated (see, e.g., Magnuson et al., 2003; McQueen, 2003; Norris et al., 2003; Samuel, 2001; Samuel & Pitt, 2003).

ASR systems do not use the kind of on-line feedback that has been the focus of so much debate in the HSR literature. In part this is for the reason noted by Norris et al. (2000): Feedback cannot do anything to improve the process of matching prelexical representations onto lexical representations. Given a particular prelexical analysis, optimal recognition is achieved simply by selecting the word representation that best matches the representation of the input. This is a formal property of pattern recognition systems in general, and so there is simply no advantage to be gained by using feedback. However, there is another reason why ASR models do not incorporate feedback between lexical and prelexical processes. As observed earlier, in mainstream systems, acoustic models are directly matched against the signal, and there is a unified search process that considers information from all levels simultaneously. Since the prelexical and lexical levels are not distinct, there is no scope for feedback between levels. Both lexical information and the language model can change path scores. If this alters the best path, then the sequence of phonemes on the best path will change, but this will have no effect on the fit of an acoustic model to a stretch of speech. In an exact parallel with the Merge model (Norris et al., 2000), lexical information can change the interpretation of the input, but cannot change the processing of the prelexical information itself.

In terms of a computational analysis of speech recognition, therefore, there appears to be no function for feedback from the lexical to prelexical levels. There is one exception, however. As Norris et al. (2003) have argued, feedback can be of benefit in retuning prelexical representations. The experiments that Norris et al. report indeed show that listeners appear to use lexical knowledge to adjust their prelexical phonetic categories. In their experiments, listeners might hear an ambiguous phoneme in a context where the lexical information indicated how that phoneme was to be interpreted. Subsequently, listeners changed the way they categorised the ambiguous phoneme in a way that was consistent with the information provided from the lexicon. This “lexically-guided” learning is of benefit to word recognition because it would improve recognition during subsequent encounters with the same speaker. That is, feedback for learning helps to solve the invariance problem by ensuring that the recognition system can dynamically adjust to new forms of variability. It is therefore critical to distinguish between on-line feedback (where the lexical level influences prelexical processing as speech is being input to the recogniser) and off-line feedback (i.e., feedback over time, for learning). Only the latter appears to be motivated by the computational analysis of the problem of speech recognition.

In ASR, various methods have been described for adapting an ASR system to the specific speech characteristics of a specific group of test speakers or to a single speaker (see Woodland, 2001, for an overview), but none yet use the lexically-guided learning seen in human listeners. The common method is to adapt the acoustic models towards the

characteristics of the voice of the test speaker. This adaptation requires some amount of speech input (in modern adaptation algorithms on the order of a few minutes, e.g., Hazen, 2000); this input is used to adapt the acoustic models such that the match between them and the test speaker's speech is improved.

Whether feedback is necessary in the speech recognition process is a computational question that applies to both HSR and ASR. However, the question of on-line feedback does not usually arise in ASR, because of the integration of the prelexical and lexical level information into one decoding structure.

### 3.2.7 Summary

We have identified a number of key problems that must be solved for successful speech recognition: the invariance problem, the real-time processing problem, the lexical embedding problem, the segmentation problem, and the out-of-vocabulary problem. Both human and machine recognisers must include algorithms that solve these problems. We have discussed the standard approaches that have been taken in both HSR and ASR to confront these problems. In almost every case there are striking parallels between the solutions adopted in HSR and ASR. In the General Discussion, we will return to the issue of how this comparison between domains may be of value in developing both HSR models and ASR systems.

First, however, we present a new model of human spoken word recognition, called SpeM (SPEech-based Model of human speech recognition; see also Scharenborg et al., 2003a, 2003b). SpeM is a new implementation of the Shortlist model (Norris, 1994) developed using ASR techniques. In contrast to existing HSR models, SpeM can recognise words from real speech input.

## 3.3 SpeM

We had several goals in developing SpeM. First, we wanted to provide a concrete demonstration of the computational parallels between HSR and ASR. If it really is the case that ASR algorithms serve the same functions as analogous HSR mechanisms, then it ought to be possible to build an HSR model using ASR components. SpeM therefore makes the links between HSR and ASR fully explicit, and serves as an illustration that a psychological model can be built using ASR techniques. Second, as Section 3.3.5 will show, the challenge of building an HSR model with ASR techniques forced us to confront how to relate the performance of the model to measures of human performance in psycholinguistic experiments. In deriving human performance measures from the model, we were able to draw further parallels between ASR and HSR. Third, it has been difficult to evaluate the Shortlist model given the unrealistic form of the input to Shortlist (see Section 3.2.2). SpeM therefore shares all assumptions made in Shortlist, but has a probabilistic/graded input rather than a discrete sequence of phonemes (which was the case in the original 1994 implementation of Shortlist). We were thus able to test whether a

version of Shortlist would be able to recognise words given acoustic input (rather than a hand-crafted symbolic description of the speech signal). We present simulations (Sections 3-4 and 3-5) showing that SpeM can indeed recognise words from real continuous speech input. The broader goal in developing SpeM is thus that the model can be used to evaluate further how a speech recogniser should work.

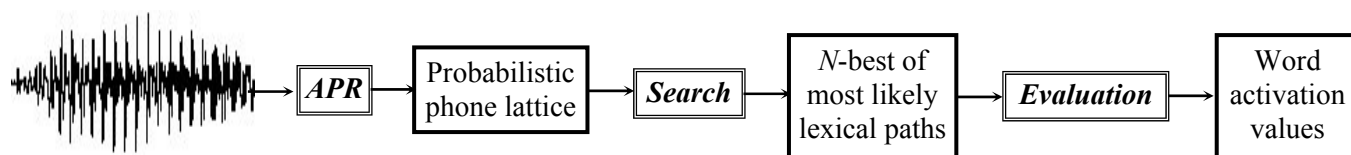


Figure 3-3. Overview of the SpeM model.

The architecture of the SpeM model is shown in Figure 3-3. SpeM consists of three modules. The first module is an automatic phone recogniser (APR) which takes the acoustic signal as its input. The APR creates a segmental representation of the acoustic signal in the form of a probabilistic phone lattice (see Section 3.3.1) using statistical acoustic models (see Section 3.2.1). This probabilistic phone lattice is then used as input to the second module, which is responsible for the lexical search. This module searches for the word (sequence) that corresponds to the best path through the probabilistic phone lattice (see Section 3.3.3) and produces output in the form of a list of the  $N$ -best paths through the phone lattice. The third module compares these alternative paths and hence computes a measure of the probability that, for a given input, individual words will be recognised (see Section 3.3.5). Each of the key computational problems identified in Section 3.2 are dealt with in the SpeM model, as described below.

### 3.3.1 Prelexical and lexical levels of processing

In Section 3.2.1, we argued that a speech recogniser must contain a mechanism to deal with the invariance problem. In HSR, it is generally assumed that this problem is solved by separating the speech recognition system into two levels, namely the prelexical and lexical levels. In many mainstream ASR approaches, these levels are intertwined in the search module by compiling grammar and lexicon into one single phone-based decoding lattice. SpeM – although based on ASR paradigms – does however consist of separate prelexical and lexical levels. In SpeM, the prelexical level is represented by the APR. The prelexical representations used in SpeM are identical to those used in Shortlist. Thus the recognition units of the APR are phones, and the probabilistic graph that will be built also consists of phones.

The APR converts the acoustic signal into a weighted probabilistic phone lattice without using lexical knowledge (see Scharenborg & Boves, 2002, for a detailed account of the APR). Figure 3-4 shows a simplified weighted phone lattice: The lattice has one root node ('B') and one end node ('E'). Each edge (i.e., connection between two nodes) carries a

phone and its bottom-up evidence in terms of negative log likelihood (its acoustic cost). The acoustic cost denotes the probability that the acoustic signal was produced given the phone ( $P(X|Ph)$ , in which  $Ph$  denotes a phone; see Section 3.2.3). The lower the acoustic cost, the more certain the APR is that the phone was indeed produced. The acoustic scores for a phone typically range from 10 to 120. For the sake of clarity, not all phones and acoustic costs are shown. Only the most probable nodes and edges for the input [as] (the Dutch word *as*, “ash”) are shown.

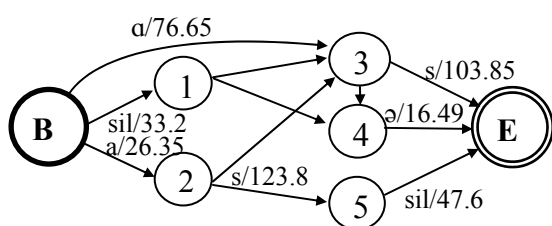


Figure 3-4. A graphical representation of a weighted probabilistic input phone lattice. For the sake of clarity, not all phones and acoustic costs are shown.

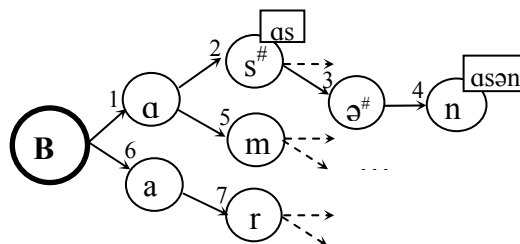


Figure 3-5. A graphical representation of the beginning of a lexical tree.

The lexical level in SpeM, as in Shortlist, has two components: the search module and the evaluation module. In the search module, one or more phonemic representations are available for each item in the lexicon. Internally, the lexicon is represented as a lexical tree in which the entries (words) share common prefix phone strings (a word-initial cohort), and each path through the tree represents a word. See Figure 3-5 for a graphical representation of the beginning of a lexical tree. The lexical tree has one root node (‘B’) and as many end nodes as there are words in the lexicon. The hash ‘#’ indicates the end of a word; the phonemic transcription in the box is the phonemic representation of the finished word. Each node in the lexical tree represents a word-initial cohort. The phonemic transcriptions belonging to the word-initial cohorts are not explicitly shown. Note that the word [as] is an example of an embedded word, since the node labelled with [as] in the lexical tree (Figure 3-5, node 2) has outgoing arcs (thus in this case the phonemic transcription [as] also represents a word-initial cohort). As described in more detail below, during word recognition the lexical tree is used to find the best paths through the phone lattice, and these paths are then evaluated relative to each other by the evaluation module.

At the lexical level, it is also possible to include knowledge on the frequencies of words (unigram language model scores) and the frequency of a word given its predecessor (bigram language model scores). These components, though implemented in SpeM, are not used in the simulations described in this paper.

### 3.3.2 Cascaded prelexical level

Rapid on-line word recognition requires cascaded processing between the prelexical and lexical levels. As reviewed earlier, HSR experiments have shown that the representations at the prelexical level should be probabilistic. In the 1994 implementation of Shortlist, however, the prelexical representations were discrete phonemes (though, as Norris (in press) points out, this was not a key assumption of the theory underlying the model). In the Merge model (Norris et al., 2000), which is derived from Shortlist, output from the prelexical level is continuous and graded. SpeM is therefore implemented in such a way that the output of the APR module is probabilistic rather than categorical. With respect to real-time processing, SpeM's search module is able to perform the search in close to real time.

### 3.3.3 Multiple activation and bottom-up evaluation of words

The lexical selection and competition stage in SpeM consists of the search module, which searches for the best path through the phone lattice and the lexical tree (see also Section 3.2.3), and the evaluation module. The search module computes the bottom-up goodness-of-fit of different lexical hypotheses to the current input, while the evaluation module acts to compare those hypotheses with each other (see Section 3.3.5). During the search process, the best path (the optimal sequence of words) is derived using a time-synchronous Viterbi search through a search space which is defined as the product of the lexical tree and the probabilistic phone lattice. In a Viterbi search, all nodes of the phone lattice are processed from left-to-right, and all hypotheses are considered simultaneously (see also Sections 3.2.1 and 3.2.3). As noted earlier, Viterbi search is simply an efficient method for finding the best path through a lattice.

The words hypothesised by the search module are each assigned a *score* (referred to as *total cost* hereafter) that corresponds to the degree of match of the word to the current input. Whenever the mismatch between the hypothesised word and the input becomes too large, the hypothesis drops out of the *beam*, that is, it is pruned away, as in ASR systems. Only the most plausible paths are therefore considered (see also Section 3.2.3).

When a node in the lexical tree is accessed, all words in the corresponding word-initial cohort are activated. Multiple activation of words is thus implemented (see Section 3.2.3). For instance, when node '2' (Figure 3-5) is accessed, not only is the word [as] activated but also all words that have [as] as their word-initial cohort.

The total cost of a path is defined as the accumulation along the path arcs of the bottom-up acoustic cost, the symbolic phone matching cost, the PWC cost, the history cost, and the word entrance penalty.

- *Bottom-up acoustic cost*: this cost is the negative log likelihood as calculated by the APR (see Section 3.3.1); it is the probability that the acoustic signal is produced given the phone ( $P(X|Ph)$ , in which  $Ph$  denotes a phone, see Section 3.2.3).
- *Symbolic phone matching cost*: this is the cost associated with the current match between the phone in the phone graph and that in the lexical tree. If the phones are identical, there are no additional costs involved. In the case of a substitution, deletion, or insertion, associated costs are added to the path costs. The associated costs for a substitution, deletion, or insertion are tuned separately.
- *PWC cost*: this cost is described in detail in Section 3.3.4.
- *History cost*: this is the total cost of the path up to the mother node, that is, the search space node from which the current search space node originates. The mother node is the previous node in the search space (i.e., in the product lattice) and is thus not necessarily the root node ‘B’.
- *Word entrance penalty*: when the search leaves the root node ‘B’ of the lexical tree, the word entrance penalty is added to the total cost of the path.

The way the total path cost is calculated in SpeM differs from mainstream ASR systems in that ASR systems do not have an explicit cost for phone-level insertions, deletions, and substitutions. Because the search in SpeM is phone based, mismatches can arise between the phonemic representation of the input in the phone graph and the phonemic transcriptions in the lexicon. It is therefore necessary to include a mechanism which explicitly adjusts for phone-level insertions, deletions, and substitutions. In mainstream ASR, however, it is usually assumed that the search space is spanned effectively by the combination of the pronunciation variants in the system’s dictionary and the system’s language model, so that the additional overhead of modelling insertions, deletions, and substitutions on the phone-level is not necessary. Furthermore, in regular ASR there is no PWC to influence the accumulated path cost. In standard ASR systems, a weighting of the acoustic cost score with a (statistical) language model score (containing, e.g., the a priori probability of a word and the probability of occurrence of a sequence of N words) determines the entire path score and therefore determines the likelihood of the path being among the ‘best paths’.

Various types of pruning (see Ney & Aubert, 1996, for an overview) are used to select the most probable hypotheses through the decoding lattice. As in Shortlist, therefore, only the most likely candidate words and paths are considered. The pruning mechanisms are:

- *Number of nodes*: A maximum number of search space nodes (320 per input node in the present simulations) are kept in memory. After each cycle of creating new search space nodes, the active nodes are sorted according to their total cost; only the top maximum number of search space nodes are kept, the rest are discarded.



- *Local score pruning*: A new search space node is only created if the total cost of the new path is less than the total cost of the best path up to that point plus a pre-set value.
- *No duplicate paths*: Of the search space nodes that represent duplicate paths, only the node with the cheapest path is kept.

The search algorithm in SpeM works as follows. The search algorithm starts in the initial search space node of the product lattice. This is denoted as (B,B), meaning that the search algorithm starts both in the root node of the phone lattice (Figure 3-4) and the root node of the lexical tree (Figure 3-5). As already indicated, the search algorithm is time-synchronous. First node '1' of the phone lattice is evaluated:

- The phone on the incoming arc of node '1' is compared with the phones in the nodes directly following the root node of the lexical tree (resulting in search space nodes (1,1) and (1,6)). If no match is found, this counts as a substitution, and the substitution cost is added to the total cost of the path; if a match is found, no costs are added.
- The phone on the incoming arc of node '1' is compared with the phones in the daughter nodes of the nodes directly following the root node of the lexical tree (resulting in search space nodes (1,2), (1,5), and (1,7)). This counts as an insertion (i.e., the insertion cost is added to the total path cost).
- The phone on the incoming arc of node '1' is compared with the phones in the root node of the lexical tree (resulting in search space nodes (1,B)). This counts as a deletion, and the deletion cost is added to the total path cost.

After all incoming arcs of node '1' of the phone lattice have been evaluated and the new search space nodes have been created, the incoming arcs of node '2' of the phone lattice are evaluated (note that in Figure 3-4, nodes '1' and '2' both have only one incoming arc, but node '3', for example, has three). In this way, paths are created through the phone lattice and the lexical tree. A path consists of a sequence of candidate words with possibly a word-initial cohort at the end of the path. Each word and word-initial cohort obtains an activation that is calculated using Bayes' Rule (see Section 3.3.5).

Let's look in more detail at path 'B-3-E' through the phone lattice compared to the path 'B-1-2' for the Dutch word *as* ([as]) through the lexical tree. The total path cost at input node '3' is the sum of the acoustic cost (which is 76.65, see the arc between the nodes 'B' and '3' in Figure 3-4), the word entrance penalty (in this case say: 50), the phone matching cost (here 0, because there is a perfect match between the phone on the arc in the phone graph and the phone in state '1' of the lexical tree), the PWC cost (here 0, because there are no insertions) and the history cost (here 0, because there is no history), thus in total: 126.65. The total path cost of this path through the phone lattice at the end node 'E' is the sum of the acoustic cost (103.85), the word entrance penalty (which is 0, because we are already in a word and not entering one), the phone matching cost (here 0, because there is a perfect match between the phone on the arc in the phone graph and the phone in state '2' of the

lexical tree), the PWC cost (here 0, because there is no phone sequence between words) and the history cost (which is now 126.65, the cheapest path to the mother node '3'), thus in total: 230.5. When comparing the word *Assen* ([ɑsən], the name of a Dutch city) with the same path through the phone lattice, the total path cost would be 230.5 plus twice the deletion cost (because both the [ə] and [n] are not to be found in the phone lattice and thus must have been deleted if this word were the source of this input). The path containing the most likely sequence of words has the highest activation (and the lowest total path score).

The output of the search module is a list of the best paths through the search space. The search algorithm thus implements multiple activation of lexical hypotheses (or sequences of words in a hypothetical path), and evaluation of each of these hypotheses with respect to the bottom-up information in the speech signal. The 'shortlist' of best paths is then input to the evaluation module. Before turning to this module, however, we first describe one further component of the bottom-up evaluation process.

### 3.3.4 Segmentation of continuous speech

In Section 3.2.5, it was argued that human listeners use a mechanism called the Possible Word Constraint for the segmentation of continuous speech into a sequence of words. The implementation of the PWC in SpeM is based on the implementation in the Shortlist model, which is that if a stretch of speech between the edge of a candidate word and the location of a likely word boundary is itself not a possible word, then that parse of the input is penalised. In SpeM, this procedure is implemented using 'garbage' symbols, comparable to the 'acoustic garbage' models in ASR systems. In such systems, garbage models are used to deal with phone insertions. A garbage model is effectively a phoneme that always has some small  $P(X|phoneme)$ . That is, it will always match the input to some degree, but will never figure in the final interpretation of the input if there is a cheaper path through the lattice consisting in a contiguous sequence of real words. The garbage symbols in SpeM match all phones with the same cost and are hypothesised whenever an insertion that is not word-internal occurs on a path. A garbage symbol (or a consecutive sequence of garbage symbols) is itself regarded as a word, so the word entrance penalty is added to the total cost of the path when garbage appears on that path.

The PWC evaluation is applied only to paths on which garbage is hypothesised. Word onsets and offsets, plus utterance onsets and offsets and pauses, count as locations relative to which the viability of each garbage symbol (or sequence of symbols) is evaluated. (Note that, as shown in Figure 3-5, the ends of words in the lexical tree are marked with a hash '#', and word onsets can be found because the mother node is 'B'.) If there is no vowel in the garbage sequence between any of these locations and a word edge, the PWC cost is added to the total cost of the path. More specifically, when the search goes through the root node of the lexical tree and the recognition of a new non-garbage word has begun, there is a PWC check on the sequence of garbage symbols.

### 3.3.5 Lexical competition and word activation

The output of the search module in SpeM is a ranked  $N$ -best list of alternative paths, each with an associated path score. This is inadequate as the output of an HSR model for two reasons. First, although the path scores reflect the goodness of fit of each path to the current input, they are not normalised relative to each other. That is, each path score is independent of all other path scores. As we discussed in Section 3.2, however, human word recognition appears to involve some kind of lexical competition, in which different lexical hypotheses are compared not only with the speech signal but also with each other. Second, the search module computes only *path*-based scores (to guide the search), not word-based scores. (The search module does have access to word scores, but does not use them to order the word sequence hypotheses). A central requirement of any HSR model is that it should be able to provide a continuous measure (usually referred to as ‘activation’ in the psychological literature) of how easy each *word* will be for participants to respond to in listening experiments. To relate the performance of SpeM to psycholinguistic data, it is therefore necessary to derive a measure of ‘word activation’ from the path scores. These two functions, relative ranking and evaluation, are provided by the evaluation module.

The way SpeM computes word activation is based on the idea that word activation is a measure related to the bottom-up evidence of a word given the acoustic signal: If there is evidence for the word in the acoustic signal, the word should be activated. The second set of factors that are relevant for the computation of word activation are the scores of the complete paths (hypotheses of word sequences) in the  $N$ -best lists.

Obviously, the total score of a path (i.e., the score of the path starting at the initial node of the lattice up to the last node of the path under construction) does not give us a direct estimate of the activation of individual words along this path. Since the path score is computed incrementally as the input unfolds over time, the best (cheapest) path from the beginning of the utterance until a certain time  $t$  changes over time; therefore, words on the best path at one point during the input need not be on the best path at a later time. Thus, broadly speaking, for each  $t$ , the best path so far does indicate *which* words are most likely to be in the input processed so far. The current implementation of word activation in SpeM therefore applies the idea that the word activation of a word  $W$  is based both on the bottom-up acoustic score for the word  $W$  itself and the total score of the path containing  $W$ .

A possible approach to derive a measure of word activation might be to calculate the Bayesian probability of each word  $W$  in the utterance which would take into account the probability of all paths on which word  $W$  appears at the same moment in time. However, although this might be possible in principle (see Norris et al., in preparation, for an example of a Bayesian approach in a much simpler HSR model), there are considerable practical difficulties in calculating such probabilities accurately with real speech input. In what follows we will develop a simplified measure of word activation which takes into

account both the bottom-up evidence for a word, and the probability of the path that the word lies on.

The word activation of a word  $W$  is closely related, in terms of Bayes' Rule, to the probability  $P(W|X)$  of observing a word  $W$ , given the signal  $X$ . Bayes' Rule and this probability play a central role in the mathematical framework on which statistical pattern matching techniques are built (i.e., most ASR implementations). Using Bayes' Rule to rank competitors is, for instance, also used by Jurafsky (1996) in his probabilistic model of lexical and syntactic access and disambiguation. The probability  $P(W|X)$  is the foundation on which we base the calculation of word activation (Scharenborg et al., 2003c).

In the current SpeM-implementation, the procedure for computing word activation of word  $W$  at time  $t$  is as follows. First, the best path that contains that word  $W$  at time  $t$  is determined. Then, the posterior probabilities for word  $W$  itself and for the best path on which  $W$  lies on the basis of the word's score (based on the acoustic score, and penalties for insertions, deletions, and substitutions) are calculated. The details on how these probabilities are computed are given in Scharenborg et al. (2003c). The key components of these computations, however, are as follows.

The probability of word  $W$  given the acoustic signal  $X$  is based on Bayes' Rule:

$$P(W | X) = \frac{P(X | W) \cdot P(W)}{P(X)}, \quad (3-1)$$

in which  $P(W)$  is the prior probability of  $W$ , and  $P(X)$  denotes the prior probability of observing the signal  $X$ .

This prior probability  $P(X)$  formally denotes the a priori probability of 'observing' the signal  $X$ . To ensure a proper normalisation of the a posteriori probability  $P(W|X)$ ,  $P(X)$  is often evaluated as follows:

$$P(X) = \sum_W P(X | W) \cdot P(W), \quad (3-2)$$

where the sum is taken over all words  $W$ . In our case, however, we do not have all this information available due to the limited length of the  $N$ -best lists that are outputted by SpeM. Instead, we evaluate  $P(X)$  as follows:

$$P(X) = D^{\#nodes} \quad (3-3)$$

In this equation,  $D$  denotes a constant (which is to be calibrated on a corpus). The exponent of  $D$ ,  $\#nodes$ , refers to the number of nodes of the path, starting from the beginning of the graph. In other words, this number refers to the number of units (phones) along that path.

The effect of this choice for  $P(X)$  is that the probability  $P(X|W) \cdot P(W)$  is normalised on the basis of the number of phones along the path that is making up the word sequence  $W$ . This normalisation is very similar to the normalisation of acoustic scores applied in the evaluation of confidence measures, the difference being that these confidence normalisations are often based on the number of frames instead of on the number of phones. This option has been chosen since, in SpeM, we do not have information about the number of frames in the input. Instead, we use the number of units (nodes) in the phone graph.

Equation 3-1 refers to a static situation, in which the signal  $X$  is specified. We are interested in how lexical activation changes over time, however. When the search process is processing the speech signal a short time after the start of  $W$ , a word-initial cohort of  $W$  (denoted  $W(n)$ , where  $n$  is the number of phones of  $W$  processed so far) will start to appear at the end of a number of hypothesised paths. Incorporating this into Equation 3-1 leads to:

$$P(W(n) | X_w(t)) = \frac{P(X_w(t) | W(n)) \cdot P(W(n))}{P(X_w(t))}, \quad (3-4)$$

where  $W(n)$  denotes a phone sequence of length  $n$ , corresponding to the word-initial cohort of  $n$  phonemes of  $W$ .  $W(5)$  may, for example, be /amstə/, i.e., the word-initial cohort of the word ‘amsterdam’. Note that  $n$  is discrete because of the segmental representation of the speech signal.  $W$  is thus a special case of  $W(n)$ : In this case,  $n$  is equal to the total number of phones of the word.  $X_w(t)$  is the gated signal  $X$  until time  $t$  (corresponding to the end of the last phone included in  $W(n)$ ).  $P(X_w(t))$  denotes the prior probability of observing the gated signal  $X_w(t)$ .  $P(W(n))$  denotes the prior probability of  $W(n)$ . In the simulations reported in this paper,  $P(W(n))$  is the same for all cohorts and all words – that is, all words and cohorts have equal a probability.

Of all the paths carrying  $W(n)$ , there will be one path with the lowest (i.e., best) overall path score (i.e., lowest at that particular moment in the search). This particular path is used to evaluate the word activation of  $W(n)$  at this point in time. The probability of this path is similar to Equation 3-4:

$$P(Path | X_p(t)) = \frac{P(X_p(t) | Path) \cdot P(Path)}{P(X_p(t))}, \quad (3-5)$$

where  $Path$  is the entire path that  $W(n)$  is on from the root node up to the time instant of interest.  $X_p(t)$  is the gated signal  $X$  until time  $t$  (corresponding to the end of the last phone included in  $Path$ ).  $P(X_p(t))$  denotes the prior probability of observing the gated signal  $X_p(t)$ .  $P(Path)$  denotes the prior probability of  $Path$ . In SpeM,  $P(Path)$  is simply the product of the prior probabilities of all words on that path, due to the fact that in all simulations are based on a unigram language model

Both Equations 3-4 and 3-5 deal with normalisation over time. The probabilities they compute are not yet normalised over paths, however. That is, these probabilities reflect the goodness of fit of each intended path/word to the input, but do not take into account the goodness of fit of other words/paths. In order to make an across-path normalised word activation measure, the multiplication of the word and path probabilities is divided by the sum of all word and path probability multiplications of all word candidates in the  $N$ -best list at a particular moment in time (this is what we refer to as the ‘probability mass’). The value of the multiplication of the word and path probabilities for a certain word is thus considered in relation to the value of the multiplications of the word and path probabilities of competitors of this word. The result of the normalisation is an activation measure that is both normalised over time and across paths.

Although an across-path normalisation is necessary, it is not necessary to use the entire probability mass that is present in the full word lattice for the normalisation. Instead, it is sufficient to normalise the multiplication of the word and path probabilities of a certain word by taking into account the multiplications of the word and path probabilities of a sufficient number of possible word candidates for a certain stretch of speech. Clearly, the longer the  $N$ -best list is on which the normalisation is based, the more robust and reliable are the results. In our case, an  $N$ -best list of 50 has proved to give robust results, in the sense that the results of the simulations reported here did not change when an  $N$ -best list longer than 50 was used.

The time and path normalised word activation ( $Act(W(n))$ ) for a word  $W$  is therefore calculated as follows:

$$Act(W(n)) = \frac{P(W(n) | X_w(t) \bullet P(Path | X_p(t))}{Pr Mass}, \quad (3-6)$$

in which  $Pr Mass$  denotes the probability mass.

We can make the following two observations about this definition of word activation. First, the activation of a word  $W$  is computed using the probability of the word  $W$  itself, and of the *best* path containing the word  $W$ , and is normalised by the sum of the word and path probability multiplications of all words in the  $N$ -best paths. An alternative approach would

be to use *all* paths containing  $W$  in the  $N$ -best list to compute the numerator in Equation 3-6. The difference between these two approaches – taking only the best path or taking (a weighted sum over) all appropriate paths – reflects the conceptual difference between ‘the winner takes all’ (i.e., neglecting entirely the presence of tokens of the word  $W$  on competing paths), and allowing several tokens of  $W$  to contribute to the overall word activation of the word  $W$ , following the assumption that the more often word  $W$  appears on a path in the  $N$ -best list the more likely it is the word  $W$  was actually produced. We chose the winner-takes-all option in SpeM because it is more transparent and easier to compute. If all paths containing  $W$  were included in the computation, a decision would have to be taken about the temporal alignment of  $W$  on different paths. That is, how closely matched in time would  $W$  have to be on different paths for those paths to contribute to the same word activation score? This issue has been addressed in ASR (see, e.g., Wessel et al., 2001), but is currently beyond the scope of SpeM. The winner-takes-all measure nevertheless tends to provide an upper estimate of word activation.

The second observation about this word activation measure is that it has close parallels with ASR *confidence* measures (e.g., Bouwman et al., 2000; Wessel et al., 2001). The confidence measure is the degree of certainty which an ASR system has that it has recognised that word correctly. Such a measure can be of value, for example, in monitoring the ongoing dialog in directory-assistance systems. The calculation of word activation in SpeM is in at least two respects similar to the calculation of word confidence measures. First, both word activation and the word confidence measure need a well-defined mapping from the (non-probabilistic) acoustic and language-model scores in the search lattice to the probabilistic domain. In SpeM, Bayes’ Rule plays a central role in this mapping. In ASR, the raw arc scores in the word graph are converted into arc probabilities. Wessel et al. use a word graph to combine the raw scores of all word instances on all paths through the search lattice and hence derive a confidence measure. Thus, although the implementation of the word activation measure in SpeM and the word confidence measure in ASR systems such as that of Wessel et al. are different, both are able to relate word-based measures with lattice-based approaches. Second, for the evaluation of the confidence measure as well as the activation measure, one must rely on certain approximations in the construction of word graphs. In a real-world ASR system, an ideal word graph is not available. Instead, the word graph is a result of choices imposed by various constraints based, for example, on numerical and memory restrictions. Wessel et al. nevertheless show that a realistically pruned word graph can be used to derive satisfactory confidence measures. In the case of SpeM, similar kinds of restrictions mean that, in standard simulations, only the 50-best paths are available at any moment in time.

In summary, the word activation measure in SpeM provides a joint measure of the goodness of fit of the word to a particular stretch of a given input and the goodness of fit of the path on which that word occurs to the complete input (more specifically, the score of the best path associated with that word). It uses Bayes’ Rule to provide an estimate of the

probability that a listener would identify that word given that input – an estimate which changes over time as the speech input unfolds.

### 3.3.6 No feedback from the lexical level to the prelexical level

In Section 3.2.6, it was argued that, during word recognition, information flows from the prelexical level to the lexical level, but not back from the lexical to the prelexical level. In SpeM, the prelexical level creates a phonemic representation of the acoustic signal, which is passed on to the lexical level. There is no top-down flow of information from the lexicon to the prelexical level. The intermediate phonemic representation of a given input at the prelexical level cannot be altered once it is created, so lexical information cannot be used on-line at the prelexical level to guide the word recognition process. This feedforward architecture is partly motivated by the parallels with Shortlist. More fundamentally, however, as we noted earlier, adding feedback would be pointless as it could not possibly improve the recognition performance of the model.

SpeM is a computational model of human *word* recognition. If one wanted to model *phoneme* recognition and, for example, lexical effects on phonetic perception with SpeM, then feedback from the lexical to the prelexical level would still not be necessary. In analogy with Merge (Norris et al., 2000), a phoneme decision layer could be added to SpeM. This layer would receive input both from the APR and the lexical evaluation modules.

### 3.3.7 Summary

In developing SpeM, we provided a concrete demonstration of the computational parallels between HSR and ASR. The solution to the invariance problem in SpeM is the separation of word recognition into three stages, an ASR-based APR at the prelexical level, and, at the lexical level, an ASR-based Viterbi search and an ASR-based evaluation procedure. The real-time processing problem is addressed using probabilistic output from the APR, and a time-synchronous lexical search that performs close to real time. The search and evaluation procedures also provide solutions to the lexical embedding problem (since all matching candidate words in the lexical tree are considered in parallel during the search and then compared during evaluation) and the segmentation problem (since selection of the best paths through the search space entails segmentation of continuous speech into word sequences even in the absence of any word boundary cues in the speech signal). Finally, the implementation of the PWC cost in the search process offers a solution to the out-of-vocabulary problem. The PWC cost penalises paths that include impossible words (garbage sequences without vowels), but does not penalise those with garbage sequences which do contain vowels. Such sequences are potential novel words. SpeM therefore offers algorithmic solutions for all of the computational problems in spoken word recognition that were discussed in Section 3.2. An obvious question now arises: How does SpeM perform?



### 3.4 Recognition of words given real speech input

Our first set of simulations sought to answer the most fundamental question that can be asked about SpeM's performance: How well can the model recognise words given an acoustic speech signal as input? We addressed this question by comparing the performance of SpeM on the recognition of a large sample of Dutch words taken from a multi-speaker corpus of spontaneous speech recorded in natural circumstances (thus including background noise), with the performance of the Shortlist model on the same materials. If our computational analysis of speech recognition is accurate, then, since SpeM instantiates algorithms to deal with the principle problems of spoken word recognition, it ought to be able to perform this task reasonably well. Note that the model need not perform perfectly for one to be able to conclude that the assumptions made by the model are justified; good performance in the recognition of words from a large vocabulary, spoken by multiple speakers and recorded in natural circumstances, and on the basis of the acoustic signal, rather than an idealised transcription of speech, already goes far beyond what any previous model of human spoken word recognition has achieved.

The comparison of SpeM with Shortlist allowed us to test the effectiveness, in terms of word recognition accuracy, of the principle difference between the two models. Unlike the original implementation of Shortlist (which we used here), SpeM has a probabilistic rather than a categorical prelexical level. As we argued earlier, probabilistic prelexical processing should provide SpeM with more flexibility to deal with the variability in the speech signal (the invariance problem, Section 3.2.1). In particular, if there is some degree of phonetic mismatch between the current input and stored lexical knowledge, as would be expected in multiple-speaker and noisy background testing conditions, a model with probabilistic prelexical output ought to be able to recover more readily than one with categorical prelexical output. Consider, for example, an ambiguous input /?it/, as a token of the word *seat*, where /?/ is ambiguous between /s/ and /ʃ/. If a categorical prelexical level decided that /?/ was /ʃ/, then recovery of the intended word *seat*, and rejection of the competitor *sheet*, would be more difficult than in a model where varying degrees of support for both /s/ and /ʃ/ could be passed up to the lexical level. Note that there are in fact two inter-related reasons why a probabilistic prelexical level should perform better than a categorical level. First, multiple phones can be considered in parallel in the probabilistic system. Second, those phones can be differentially weighted, as a function of their degree of match to the input. If SpeM were therefore to perform better than Shortlist on the materials from the Dutch spontaneous speech corpus, then this would reflect the increased flexibility and robustness of word recognition provided by a probabilistic prelexical level.

In this first set of simulations, we also examined another aspect of the invariance problem. As we described in Section 3.2.1, due to the variation found in everyday speech, the number of phonemes in a word that are actually produced may differ from the number of phonemes in the canonical representation of that word (either because of phone insertions

and/or because of phone deletions). Furthermore, the identity of the phonemes themselves may vary too (because of phone substitutions). We therefore also addressed how a speech recogniser should deal with the fact that real speech often does not align segmentally with predefined lexical representations.

At the lexical level in SpeM, each word has a representation that includes an abstract specification of its phonological form, specifically, a sequence of phones in the lexical tree (see Figure 3-5). The lexical representations in Shortlist are also sequences of phonemes. It might therefore appear that both of these models would be unable to recognise words that had undergone phone insertions or phone deletions. There are two features of SpeM, however, that might allow it to deal with this problem. First, SpeM does not use simple lexical look-up (as Shortlist does). Instead, it uses a DP algorithm that is able to align two strings of different lengths (see Section 3.2.3). This means that when a phone insertion occurs, for example, the mismatch with lexical representations need not be so severe in SpeM as in Shortlist. In particular, the insertion would not cause all subsequent phones in the input to be misaligned, as occurs in Shortlist. Second, SpeM includes insertion and deletion scores (see Section 3.3.3). In the context of a DP algorithm, which tolerates misalignment between the input and canonical lexical representations, it is necessary to include a mechanism which acts to rank the relative goodness of fit of different degrees of mismatch. For example, an input with one phone insertion relative to a canonical pronunciation of a word ought to be a better match to that word than to another word where the difference entails two or more insertions. The insertion and deletion costs in SpeM provide this mechanism. In the following set of simulations, we examined whether the DP algorithm, modulated by the insertion and deletion scores, would allow SpeM to recognise words in spite of insertions and deletions. We compared a version of the model with the insertion/deletion penalties (see Section 3.3.3) set so high that the model did not tolerate any insertions or deletions in the input (SpeM-I/D) with one in which the scores were set at normal levels (SpeM+I/D/S).

We also examined the effect of SpeM's substitution penalty by including a simulation run in which not only the insertion and deletion penalties were set very high, but also the substitution penalty was set such that the model did not tolerate any substitutions in the input (SpeM-I/D/S). Finally, we investigated whether there were any differences in performance levels between Shortlist and SpeM as a function of the different types of lexical search in the two models (a DP technique in SpeM; a lexical look-up procedure in Shortlist). Both models were presented with categorical input: the first-best phoneme string as output by the APR (we refer to this version of SpeM as SpeM-cat; note that the insertion, deletion, and substitution penalties in this simulation were the same as in the SpeM+I/D/S simulation).

### 3.4.1 Method

The APR (consisting of 36 context-independent acoustic phone models, one silence model, one model for filled pauses, and one noise model) was trained on 24,559 utterances taken from the Dutch Directory Assistance Corpus (DDAC; Sturm et al., 2000). Each utterance consisted of a Dutch city name pronounced in isolation. The same APR was then used for the Shortlist simulation and for the SpeM simulations. The outputs of the APR were probabilistic in the SpeM+I/D/S, SpeM-I/D, and SpeM-I/D/S simulations (i.e., they took the form of a probabilistic phone graph, see Section 3.3.1). Because Shortlist takes a symbolic description of the speech signal as input, it is not able to recognise words given real speech input. The APR-module of SpeM was therefore used to generate a categorical phonemic representation of the speech signal for use in the Shortlist simulation (and the SpeM-cat simulation). In both of these cases, the sequence of best-matching phones, as computed by the APR, was selected for each input.

The systems were tested on 10,509 utterances from the DDAC corpus that had not been used for training the APR. These utterances contain either a Dutch city name, the name of a Dutch province, or the Dutch sentence *ik weet het niet* ('I don't know'). The lexicon in all three simulations consisted of 2,398 entries: city names, Dutch province names, and *ik weet het niet*. For each entry in the lexicon, one unique canonical phonemic representation was available. Prior to the test, all models were optimised on a subset of 100 utterances from this test corpus. Parameter values in both Shortlist and SpeM were adjusted to maximise the number of correctly recognised words in each case. In Shortlist, the optimised parameter was the mismatch parameter (see Scharenborg et al., 2003d, also for related Shortlist simulations).

### 3.4.2 Results and discussion

Performance of the Shortlist and SpeM models was evaluated using the ASR benchmarking method of recognition performance. Recognition performance was therefore measured in terms of word accuracy: The percentage of utterances for which the word in the orthographic transcription of the test material received the highest activation value in the output of Shortlist or SpeM.

The results are presented in Table 3-1. There are four key aspects to these findings. First, the comparison of the performance of Shortlist and SpeM-cat shows that the lexical search as implemented in SpeM is better able to match the input string onto lexical items. The 3.5% gain in performance is solely contributable to the implementation of the search since the earlier components of the two systems were kept the same (i.e., the same APR producing the same phoneme strings). This shows that the DP implementation in SpeM is somewhat better able to deal with the variability in real speech materials than the lexical look-up process in Shortlist. In particular, the DP algorithm provides more flexibility in dealing with insertions, deletions, and substitutions. It is important to note that the

mismatch parameter in Shortlist provides some tolerance for phone substitutions: If this parameter is not set too high, words can still be recognised in spite of a mismatch between the input and that word's canonical representation. In the present simulations, however, the mismatch parameter was adjusted during optimisation. Even though Shortlist was therefore operating with an optimised mismatch parameter, it appears that the DP search algorithm in SpeM works somewhat better in dealing with non-canonical input.

*Table 3-1.* Results on the Dutch Directory Assistance Corpus test utterances for Shortlist and four versions of SpeM, one in which the APR produced categorical phonemic output (SpeM-cat), and three in which it produced probabilistic output: one in which phone insertions, deletions, and substitutions were tolerated by the model (SpeM+I/D/S), one in which substitutions but not insertions and deletions were tolerated (SpeM-I/D), and one in which neither substitutions nor insertions/deletions were tolerated (SpeM-I/D/S).

<b>Model</b>	<b>Accuracy (%)</b>
Shortlist	32.5
SpeM-cat	36.0
SpeM+I/D/S	72.1
SpeM-I/D	70.3
SpeM-I/D/S	64.3

Second, the difference in effectiveness of a categorical prelexical level and a probabilistic prelexical level is clearly illustrated by the comparison of SpeM-cat with SpeM+I/D/S (remember that the parameter settings in the two versions of SpeM were otherwise identical across these two simulation runs). As Table 3-1 shows, a gain of 36.1% in performance is obtained once the input has changed from a categorical sequence of phonemes to a probabilistic phone graph. SpeM+I/D/S is thus much more able than SpeM-cat to deal with the variability in real speech input. The probabilistic prelexical level of SpeM+I/D/S outperforms the categorical prelexical level of SpeM-cat (and Shortlist) because it allows the lexical search process to consider multiple phones in parallel, each with a graded degree of bottom-up support, while SpeM-cat and Shortlist only have available the most likely phone. This means that word recognition, in particular given the variability in the test materials used here, is more robust and flexible in SpeM+I/D/S (the standard version of SpeM) than in SpeM-cat and Shortlist. This finding thus supports the claim made in Section 3.2.2 that the intermediate representation of the speech signal at the prelexical level should be probabilistic rather than categorical.

Third, the analyses of the benefits of the insertion, deletion, and substitution penalties show that although all three mechanisms improve recognition accuracy, tolerance of phone substitutions is more important than tolerance of insertions and deletions. The comparison of the performance of SpeM+I/D/S and SpeM-I/D/S shows that the joint mechanisms of a DP search algorithm and the insertion/deletion/substitution costs help the model to recognise words when the input mismatches with canonical lexical pronunciations. Recognition accuracy improved by 7.8% when the insertion, deletion, and substitution costs were set at a level which allowed the DP algorithm to find lexical matches in spite of phone mismatches. The bulk of this benefit is due to the effect of the substitution costs. When the substitution penalty was operating normally, but the insertion and deletion penalties were very high (the SpeM-I/D simulation), there was only a 1.8% change in recognition performance.

Fourth, the recognition rate of the standard version of the SpeM model (SpeM+I/D/S) is 72.1%. This means that SpeM can recognise over two thirds of all words in the test corpus of (mainly) isolated words, spoken in a spontaneous speech setting (a directory assistance system) by a variety of speakers. No previous HSR model has done this. This is made clear by the SpeM+I/D/S – Shortlist comparison: SpeM performed more than twice as well as Shortlist.

Would the performance of Shortlist have been better had we used a narrow transcription of the speech signal created by a human transcriber rather than the APR? In Scharenborg et al. (2003d), we argue that this would not have been the case. Cucchiaroni et al. (2001) showed that automatically generated transcriptions of read speech are very similar to manual phonetic transcriptions created by expert phoneticians. Such human transcriptions are to a large extent also non-canonical, just as the transcriptions created by the APR. Thus, we would predict that input created by human expert transcribers would result in a similar level of recognition performance in Shortlist.

One might also ask how SpeM would compare with conventional ASR systems on the same recognition task. In Scharenborg et al. (2003b), SpeM was not only compared with Shortlist but also with an off-the-shelf ASR system. The performance of SpeM fell short of the performance of the ASR system. Using the same lexicon, the ASR system reached an accuracy of 84.9%. This might in turn raise the question: Why not use this ASR system as a model of HSR instead of SpeM? This would not be appropriate, however, since ASR systems are not designed for the simulation of human word recognition processes nor must the design choices in such models respect the available data on HSR. In short, ASR systems are not models of HSR. Scharenborg et al. (2003b) suggest that the poorer performance of SpeM was attributable to two factors: the limitations of the APR used in SpeM, and the more complex lexical search algorithms used in the ASR system. It may be possible to improve SpeM's performance by enriching the DP technique that is currently employed. There is, however, no psychological data that would support any such specific

adjustments, and, more fundamentally, we doubt whether such an attempt to improve SpeM's recognition performance by a few percentage points would lead to any further understanding of HSR.

We hope that in the future it will be possible to improve the APR module in SpeM. This will be an important issue to pursue since the computational analysis (Section 3.2.1) suggests that an effective prelexical level is essential for large-vocabulary speaker-independent word recognition.

### 3.5 Recognition of words in continuous speech

#### 3.5.1 Temporarily lexically ambiguous input

The simulations reported in Section 3.4 show that SpeM is able to recognise over two thirds of the words in real, spontaneous speech, where the input mainly consisted of isolated words. In this section, SpeM's ability to recognise words in continuous speech will be addressed. We took the approach in the next simulation of examining the performance of the model on a specific input, rather than on a large corpus of materials (as in the preceding simulations). Thus, instead of using global recognition accuracy measures, we focussed on SpeM's performance at the item-specific level. This level of analysis provides valuable insights into the detailed working of the model. SpeM was confronted with input that was temporarily lexically ambiguous: the utterance *ship inquiry*. Such utterances can effectively 'garden-path' a listener or recogniser. After [ʃɪpɪŋ] the input matches the word *shipping*, and this may be the preferred analysis of the input. However, the only word that matches the final three syllables of the utterance is *inquiry*. At the end of the utterance, therefore, the only fully consistent parse of the input is *ship inquiry* and the initial analysis of the input must be revised. This example was used by Norris (1994) to show how the relative evaluation of material in non-overlapping portions of the input in the lexical competition process in Shortlist can allow the model to derive the correct interpretation of this input. The example thus provided an excellent test of the ability of Shortlist to select the optimal interpretation of an input sequence which was temporarily lexically ambiguous, and which initially offered more bottom-up support for an incorrect word (i.e., more support for *shipping* than for *ship*). In this simulation, therefore, we tested whether SpeM would also be able to segment the continuous input [ʃɪpɪŋkwærɪ] into the correct sequence of words.

#### *Method and material*

First, the APR component of SpeM was trained on British English. Forty four acoustic phone models, one silence model, and two noise models were trained on 35,738 British English utterances taken from the Speechdat English database (Höge et al., 1999). Each utterance in the training corpus contained maximally two words.

At test, SpeM was asked to recognise *ship inquiry*. Three carefully spoken tokens of *ship inquiry* were produced by a male native speaker of British English, and recorded in a soundproof booth. The APR module converted the acoustics of each of these three recordings into probabilistic phone graphs. Subsequently, these phone graphs were fed into the lexical search module. The lexicon used in the search was identical to that used in the Shortlist simulations in Norris et al. (1997). Each word had one canonical phonemic representation, and there were a total of 26,449 lexical entries. The parameters of the model were not optimised for these specific inputs. Instead, we selected the same parameters as were optimised in previous related simulations on *ship inquiry* (simulations in which a linear sequence of phones rather than real speech was used as input; Scharenborg et al., 2003a). In addition to the word activation values generated by SpeM, we also examined the 10 best paths generated by the search module.

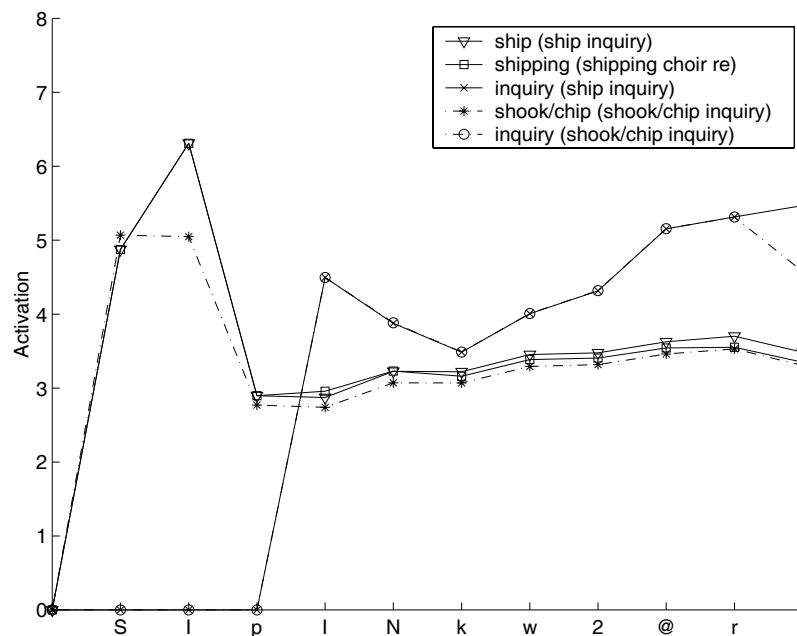


Figure 3-6. The mean word activation flows for three recordings of ‘ship inquiry’. The y-axis in all panels displays the word activation; the x-axis shows the phonemes in DISC-format (Burnage, 1990) in the input, as they arrive over time.

### Results and discussion

Appendix A shows the 10-best lists for each of the recordings. Figure 3-6 shows the average word activations of *ship* and *inquiry* (i.e., the words associated with correct recognition), *shipping* (the word also embedded in the signal), and the closest competitors (*shook* in the first recording and *chip* in the second and third recording). The path on which the word lies is shown between brackets in the legend.

For the first recording, the most likely segmentation is *shook inquiry*, while the segmentation *ship inquiry* can be found at rank 3 (see Appendix A). For the second recording, the most likely segmentation is *chip inquiry*, while the segmentation *ship inquiry* can be found at rank 2. Finally, for the third recording, SpeM is able to correctly parse the input: *ship inquiry* can be found at the first position.

The word activation of *shipping* is higher than the word activations of *ship*, *shook*, and *chip* around the phonemes [ɪ] and [ŋ], as is to be expected on the basis of the bottom-up evidence. The difference, however, is only small. This small difference is due to the small difference in the bottom-up acoustics costs associated with the phonemes in *shipping*, *chip*, *shook*, and *ship* as calculated by the APR. Towards the end of the input, the average word activation function of the parse *ship inquiry* is higher than the average word activation function of its closest competitors.

What is striking in these results is the success of the word *inquiry*. Hardly ever is *inquiry* substituted by another word. The search procedure in SpeM is thus able to select this word in 22 out of the 30 best paths across the three simulation runs, and to select this word on all three of the first best paths in each set of 10. The word activation of *inquiry* is therefore consistently high, and the word activation of the incorrect lexical hypothesis *shipping* is consistently lower. Thus, even on the inputs for which *ship* is not on the first best path, *ship* always falls on a more highly-ranked path than *shipping* does, and always has a higher word activation at the end of the input than *shipping* does. It is quite remarkable that SpeM performs this well, since the task it is faced with is not trivial. Specifically, even though the same speaker produced the same word sequence three times in the same recording environment, the three inputs generated three different APR outputs. This is of course another form of the invariance problem. In the *N*-best phone lists generated on the basis of the probabilistic phone graph created by the APR, the correct phone sequence [ʃɪpɪŋkwɪəri] was hardly ever found. The phone sequences found in these lists instead included [dʒɪtɪŋgwɪəri], [dʒɪtɪŋkwɪəri], dʒɪtɪŋkwɪəri], and [dʒʊdɪŋkwɪəri]. Furthermore, there were, on average, 3.03, 4.69, and 3.79 different phonemes (for the first, second, and third recording respectively) on parallel arcs in the phone graph. That is, SpeM had to consider, on average, more than three phoneme alternatives at any moment in time. The limitations of the APR are thus the primary reason why words such as *shook* and *chip* end up on high-scoring paths and have high word activations. In spite of these limitations, however, the search process is powerful enough for the correct lexical interpretation of [ʃɪpɪŋkwɪəri] to tend to win out. That is, SpeM is able to find the correct segmentation of this continuous real speech input, albeit not always on the first best path. For these simulations, an *N*-best list of 50 was used to calculate the probability mass. Increasing the length of the list did not influence the pattern of performance of SpeM.



### 3.5.2 Lexical competition in spoken word recognition

In Section 3.2.3, we explained that any speech fragment is likely to be compatible with many lexical alternatives, and that there is considerable HSR evidence that multiple candidates are indeed activated. In McQueen et al. (1994), for instance, human listeners were confronted with speech fragments that were either the beginning of an existing word or the beginning of a nonword. They were asked to press a button as quickly as possible if the stimulus began or ended with a real word and then say the word they had spotted aloud. The results showed that words were spotted faster, and less errors were made, if the real word was embedded in a stimulus that was not the onset of another word. This indicates that when the stimulus was the onset of an existing word that particular word was also activated, resulting in an inhibitory effect on the target word.

This competition process is implemented in SpeM's evaluation module. We have already seen how this process helps in the resolution of lexical ambiguities such as *ship inquiry*. In this section, SpeM's ability to spot words in ambiguous speech fragments is addressed further. We took the approach of examining the performance of the model on recordings of an entire set of stimuli from an HSR experiment. The test material consisted of the stimuli from the experiments described in McQueen et al. (1994). We therefore employed a third style of simulation. Rather than testing SpeM on utterances from a speech corpus (Section 3.4) or on one specific two-word sequence (Section 3.5.1), we used the complete stimulus set from a psycholinguistic experiment. This illustrates the flexibility that SpeM has as a tool for examining HSR.

SpeM was confronted with bisyllabic stimuli of which either the first or the second syllable was the target word. The full stimulus was either the start of an actual word or a nonword. In the case where the stimulus was the start of an actual word (the so-called 'embedding word'), both the target word and the embedding word should be activated, resulting in a competition effect relative to the case where the stimulus was not the onset of an actual word. Is SpeM able to simulate this effect?

#### *Method and materials*

All items (target words embedded as second syllable of Weak-Strong (WS) word onsets, WS nonword onsets, words embedded as first syllable of SW word onsets, and SW nonword onsets) used in the McQueen et al. (1994) experiment were twice carefully reproduced by the same British English speaker as in the previous simulation, and recorded in a soundproof booth. There was a total of 144 items, divided into four types of stimuli. Table 3-2 gives an example of each of the four stimulus types. (For a full list of the materials, see McQueen et al., 1994.) We preferred to use the same speaker throughout all simulations, so that the mismatch between the speaker's voice and the acoustic model set of the APR was identical across simulations.

Table 3-2. The four types of stimuli from McQueen et al. (1994).

	Words embedded as second syllable of WS words			Words embedded as first syllable of SW words		
	<i>stimulus</i>	<i>target</i>	<i>embedding word</i>	<i>stimulus</i>	<i>target</i>	<i>embedding word</i>
<b>Word onset</b>	<u>domes</u>	<u>mess</u>	domestic	<u>sacrif</u>	<u>sack</u>	sacrifice
<b>Nonword onset</b>	<u>nemess</u>	<u>mess</u>	--	<u>sackrek</u>	<u>sack</u>	--

At test, SpeM was asked to recognise the recorded items. The APR module converted the acoustics of each of the recordings into probabilistic phone graphs. Subsequently, these phone graphs were fed into the lexical search module. The lexicon used in the search was identical to that used in the Shortlist simulations in Norris et al. (1997). Each word had one canonical phonemic representation, and there were a total of 26,449 lexical entries.

#### Results and discussion

The word activations of the (cohorts of the) target words as they grow over time were extracted from the 50-best lists. For each of the four conditions, the average word activation functions are plotted. Figure 3-7 shows the activation flows of the target and the embedded words in the four conditions. The upper panel shows the activation flows for the WS stimuli; the lower panel shows the activation flows for the SW stimuli. The y-axis displays the average word activation. The nodes on the x-axis correspond to the number of input phones processed. In the upper panel, position '1' is aligned with the start of the embedding word (e.g., of *domestic*); position '3' is aligned with the start of the target word (e.g., of *mess*). The WS stimuli are such that the target word always starts at the third phoneme of the embedding word. In the lower panel, position '1' is aligned with the start of the target word (e.g., of *sack*, and thus also the embedding word, e.g., of *sacrifice*). Note, however, that since the nodes on the x-axis correspond to the number of nodes in the output graph of the APR, they thus may reflect phones which overlap partially in time. They do however obey chronological ordering: if  $m > n$ , node  $m$  has a later time stamp than node  $n$ .

McQueen et al. (1994) found that target words embedded as the second syllable of WS word onsets (e.g., *mess* in *domes*) were harder to identify than words embedded as the second syllable of WS nonword onsets (e.g., *mess* in *nemess*). Furthermore, the identification of target words embedded as the first syllable of SW word onsets (e.g., *sack* in *sacrif*) was more difficult than the identification of target words embedded as the first syllable of SW nonword onsets (e.g., *sack* in *sackrek*) after the offset of the target word. Figure 3-7 shows that SpeM is able to simulate these results. In the case of the WS syllable stimuli, the activation of the embedding word in the matched word onset situation (e.g., *domestic* in *domes*) is much higher than the activation of that word in the nonmatched word

onset situation (e.g., *domestic* in *nemess*), because there is more evidence for *domestic* in the acoustic signal in the former stimulus type. The inhibitory effect of the embedding words on the target words in the matched word onset case is larger than in the nonmatched word onset case, resulting in a higher activation for the target word in the nonmatched word onset than the matched word onset case. The lower panel shows a similar picture: The activation of the embedding word in the matched word onset case (e.g., *sacrifice* in *sacrif*) is higher than the activation of the embedding word in the nonmatched word onset case (e.g., *sacrifice* in *sackrek*). This higher activation again causes the activation of the target word in the matched word onset case (e.g., *sack* in *sacrif*) to be lower than the activation of the target word in the nonmatched word onset case (e.g., *sack* in *sackrek*) due to the larger inhibitory effect.

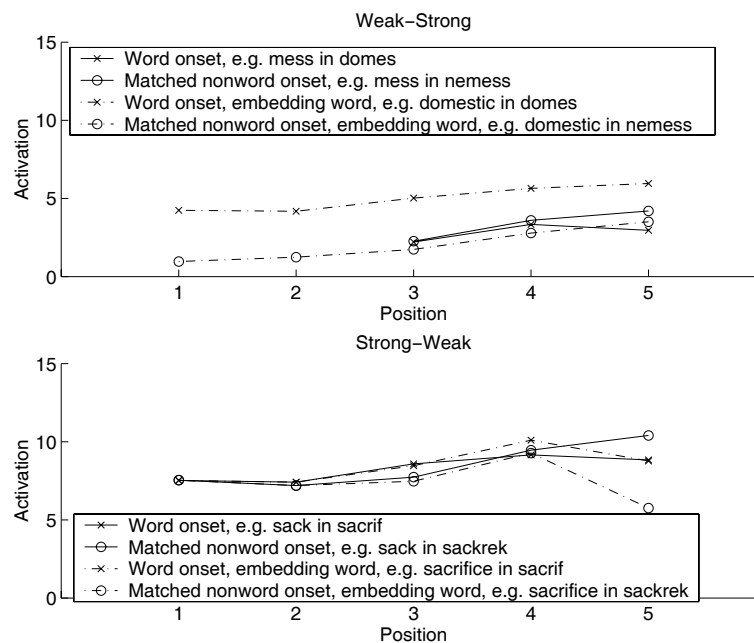


Figure 3-7. Mean activation levels for the materials in the *domes*-simulation for the four stimuli types. In the upper panel, position '1' is aligned with the start of the embedding word (e.g., of *domestic*); position '3' is aligned with the start of the target word (e.g., of *mess*). In the lower panel, position '1' is aligned with the start of the target word (e.g., of *sack*, and thus also the embedding word, e.g., of *sacrifice*).

McQueen et al. (1994) also found that the competition effect was stronger for target words that were embedded as second syllables (the WS stimuli) than for target words that were embedded as first syllables (the SW stimuli). This effect is illustrated in Figure 3-7 by the greater absolute difference in mean activations of the target words in the matched and the nonmatched word onsets in the WS stimuli (positions '3' to '5' in upper panel of Figure 3-7) versus the absolute difference in mean activations of the target words in the matched and

the nonmatched word onsets in the SW stimuli (positions ‘1’ to ‘3’ in lower panel of Figure 3-7). The earlier activation of the longer embedding word in the case of the WS stimuli causes more inhibition sooner and hence a larger competition effect at the offsets of the target words.

These results show that SpeM is able to simulate the results of the McQueen et al. (1994) experiments. The present simulations also show that SpeM can be used to simulate performance in specific psycholinguistic experiments: Recordings of the entire set of stimuli from an experiment can be given as input to the model. Note that in these simulations, an *N*-best list of 50 was used to calculate the probability mass. Increasing the length of the list did not influence the pattern of performance of SpeM.

### 3.5.3 The Possible Word Constraint and the segmentation of continuous speech

In the final simulation, SpeM’s ability to deal with the segmentation problem was investigated further. With the *ship inquiry* simulation we have seen that the lexical search procedure in SpeM can be biased by late-arriving information, such that an earlier incorrect interpretation (i.e., the word *shipping*) can be revised in favour of the correct parse of the input. That is, the lexical search and evaluation procedure is able to settle on an optimal segmentation of continuous speech input in the absence of any cues to a word boundary in that input. In the *domes* simulation we saw in addition how lexical competition between words beginning at different points in the signal can influence recognition performance across a set of stimuli from a psycholinguistic experiment, but again in the absence of any disambiguating word boundary cues. In Section 3.2.5, however, we argued that human listeners do use cues to word boundaries in the speech signal, when those cues are available, and that they do so by using a lexical viability constraint, the PWC. A word is disfavoured in the recognition process if it is misaligned with a likely word boundary, that is, if an impossible word (a vowelless sequence) spans the stretch of speech between the boundary and the beginning (or end) of that word (Norris et al., 1997). We have argued that the PWC helps the speech recogniser solve the segmentation problem and the out-of-vocabulary problem. SpeM, like Shortlist (Norris et al., 1997), therefore contains an implementation of the PWC. It was important to test whether the implementation in SpeM allows the model to simulate experimental evidence on the PWC.

SpeM was therefore confronted with words which were preceded or followed by a sequence of phones that could or could not be a possible word in English. The test material consisted of the stimuli (words embedded in nonsense words) from the PWC experiments (Norris et al., 1997). Again, we used the complete stimulus set from a psycholinguistic experiment for testing SpeM.

In the Norris et al. (1997) experiments, English listeners had to spot real English words embedded in nonsense sequences (e.g., *apple* in *fapple* and *vuffapple*). In line with the predictions of the PWC, the listeners found it much harder to spot target words when the

stretch of speech between the beginning of the target and the preceding silence was an impossible English word (e.g., the single consonant *f* in *fapple*) than when this stretch of speech was a possible (but non-existing) English word (e.g., the syllable *vuff* in *vuffapple*). Can SpeM simulate this result?

### *Method and materials*

All items (target words preceded or followed by phone sequences that are either impossible or possible words of English) used in the Norris et al. (1997) PWC experiments were carefully reproduced by the same British English speaker as in the previous simulations, and recorded in a soundproof booth. There was a total of 384 items, divided into eight types of stimuli – target words embedded in nonsense words. Table 3-3 gives an example of each of the eight stimulus types. For a full list of the materials, see Norris et al. (1997).

*Table 3-3.* The eight types of stimuli (words embedded in nonsense words) from Norris et al. (1997).

<b>Residue</b>	<b>Monosyllabic words</b>		<b>Bisyllabic words</b>	
	<i>Preceding context</i>	<i>Following context</i>	<i>Preceding context</i>	<i>Following context</i>
<b>Impossible</b>	<u>f</u> egg	se <u>sh</u>	<u>f</u> apple	sugar <u>th</u>
<b>Possible</b>	<u>ma</u> ffegg	se <u>sh</u> ub	<u>vuff</u> apple	sugar <u>thim</u>

To test whether the implementation of the PWC in SpeM allows the model to simulate experimental evidence on the PWC, the word activation flows as they grow over time were plotted for each of the eight conditions for the case where the PWC mechanism was disabled (control condition) and for the case where the PWC mechanism was enabled (following Figures 1 and 2 in Norris et al., 1997). The conditions of the simulation were otherwise identical to the previous simulation. The same APR, trained in the same way, was used. The APR converted the acoustic signal of each item into a probabilistic phone graph. Furthermore, the same lexicon as before was used for the lexical search. SpeM again calculated the 50 best paths for each of the items.

### *Results and discussion*

The word activations of the (cohorts of the) target words as they grow over time were extracted from the 50-best lists. For each of the eight conditions, the average word activation functions are plotted. Figure 3-8 shows the activation flows in the control case when the PWC mechanism is disabled; Figure 3-9 shows the activation flows when the PWC mechanism is enabled. In both figures, the y-axis displays the average word activation. The nodes on the x-axis correspond to the number of input phones processed. The activation functions are aligned relative to the first phoneme of the target word. So,

position 1 of the x-axis corresponds with the first phoneme of the target word. As was the case for the results plotted in Figure 3-7 in the previous simulation, the nodes on the x-axis correspond to the number of nodes in the output graph of the APR, and they thus may reflect phones which overlap partially in time. They do however obey chronological ordering.

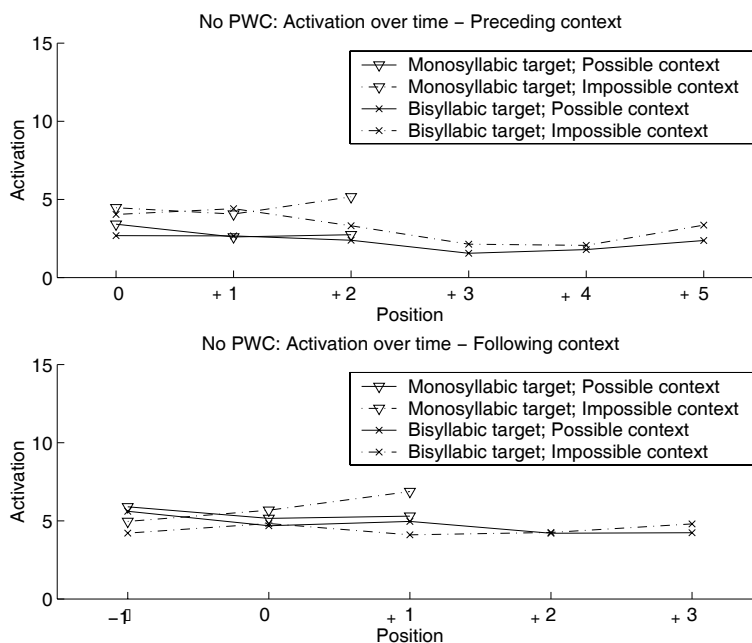
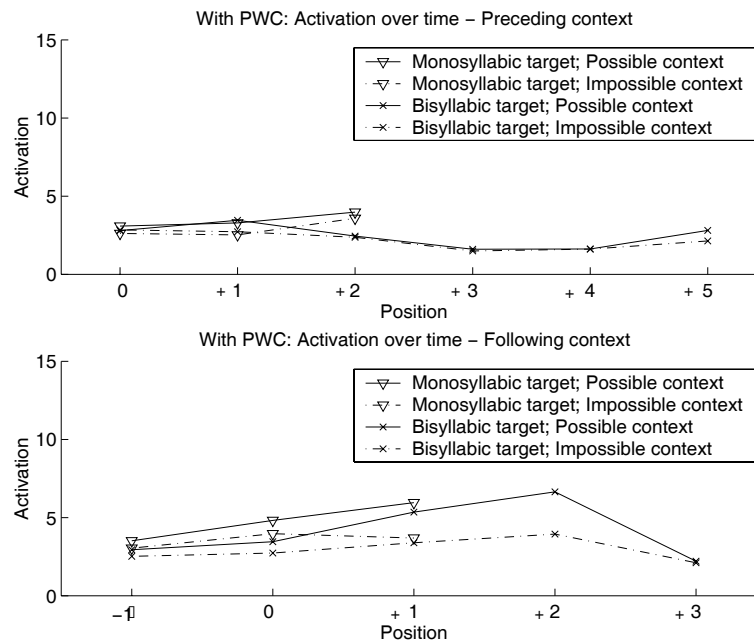


Figure 3-8. Mean target activation levels for the materials in the PWC simulation while the PWC mechanism was *disabled*. The upper panel shows the activation levels for target words with preceding context. The lower panel shows the activation levels for the target words with following context. The activation functions are aligned relative to the last/first phoneme of the target word ('0'). Thus, for targets with preceding context, '+1' is the second segment of the target word, while for targets with following context, '+1' is the first segment of the context.

Norris et al. (1997) found that words preceded or followed by residues that were possible words were more easily recognised by human subjects (resulting in faster reaction times and fewer errors) than words preceded or followed by residues that were not possible words. Figure 3-8 shows that, in the absence of the PWC, word-spotting should be harder for monosyllabic target words with possible context than for monosyllabic target words with impossible context, and in the case of preceding context that word-spotting should be harder for bisyllabic target words with possible context than for bisyllabic target words with impossible context. This is contrary to the findings reported by Norris et al. When the PWC mechanism is enabled, however, SpeM is able to correctly simulate these findings, as is shown in Figure 3-9. For each of the four different types of target word (monosyllabic or

bisyllabic, and with preceding or following context), those in possible word contexts (solid lines) have higher mean activations than those in impossible word contexts (dashed lines).



*Figure 3-9.* Mean target activation levels for the materials in the PWC simulation while the PWC mechanism was *enabled*. The upper panel shows the activation levels for target words with preceding context. The lower panel shows the activation levels for the target words with following context. The activation functions are aligned relative to the last/first phoneme of the target word ('0'). Thus, for targets with preceding context, '+1' is the second segment of the target word, while for targets with following context, '+1' is the first segment of the context.

Comparing the word activation flows in Figures 3-8 and 3-9 shows that the activations in Figure 3-9 are overall lower than the activations in Figure 3-8. This is due to the PWC mechanism. First, the addition of the PWC penalty (to target words in impossible word context) causes the word activation to be lower. Second, the addition of the PWC penalty causes more words to be absent from the 50-best list, such that there are more zero activations in the setting where the PWC mechanism was enabled, which in turn also causes the overall activations to be lower. Note also that, in these simulations, increasing the length of the *N*-best list did not influence the pattern of performance of SpeM. This shows again that the choice of a 50-best list is reasonable.

SpeM models the word-spotting data in the same way as was done by Shortlist in Norris et al. (1997): The higher a word is activated the more likely it is that that word will get a 'yes' response and the faster the response will be. With Shortlist, however, it was not possible to make a direct link between the error rates of the subjects and the error rate of the model.

The implementation of SpeM, however, makes this possible. We calculated SpeM's error rates in two different ways and compared them with the error rates of the human subjects. Table 3-4 shows the mean error rates of the human subjects ('H') and SpeM for each of the eight conditions. '1B' shows the percentage of target words that were not found on the first-best path as was calculated by SpeM; '10B' shows the percentage of target words that were not to be found in the 10-best list.

*Table 3-4.* Mean percentage error rates of the human subjects (H; taken from Norris et al., 1997, Experiment 1); of the words on the first-best path as calculated by SpeM (1B); and of the words in the 10-best list as calculated by SpeM (10B).

Residue	Monosyllabic words (%)						Bisyllabic words (%)					
	<i>Preceding context</i>			<i>Following context</i>			<i>Preceding context</i>			<i>Following context</i>		
	H	1B	10B	H	1B	10B	H	1B	10B	H	1B	10B
<b>Impossible</b>	52	79	29	39	85	6	18	98	40	38	96	17
<b>Possible</b>	57	77	19	28	69	10	14	90	40	17	92	4

Norris et al. (1997) found that responses to targets with possible context were more accurate than responses to targets with impossible context. In SpeM, for the words on the first-best path ('1B') alone, the error rates show the PWC effect in all four conditions: For each type of target word and each context position, responses to targets in possible contexts were more accurate than responses to targets in impossible contexts. Using only the first-best path to estimate error rates may be a rather strict criterion, however. Even when the error rates are calculated on the basis of the words in the 10-best list, the PWC effect is still present in 2 out of the 4 cases.

As shown in Table 3-4, however, the error data in SpeM do not align perfectly with the human error data. There are at least three reasons for this. First, as pointed out before, the APR in SpeM does not work perfectly. This certainly causes the error rates in SpeM's first-best paths to be much higher in all four conditions than those in the human data, and will also contribute to the pattern of error rates in SpeM's 10-best lists. Second, it is impossible to compare directly the error rates in SpeM's 10-best lists with the human data, since it is unlikely that humans compute specifically 10-best lists, and, even if they do, we have no access to those lists. Third, SpeM's behaviour is more deterministic than human word-spotting behaviour. While SpeM will always behave (in terms of word activation scores and error rates) in the same way on a given input, humans can show the effect of the PWC in speed or accuracy or both, and the relative weighting of these effects can vary from trial to trial and subject to subject. For all three of these reasons, we should not expect perfect correlations between the model's errors and the human error data.



Although the comparison of error rates in SpeM with the Norris et al. (1997) error data is not straightforward, it does nevertheless show that SpeM is able to model the PWC effect with real speech input not only using word activation flows but also using error rates. The PWC implementation helps SpeM to segment continuous speech fragments and to favour parses that only contain real or possible words. As we have discussed earlier, the PWC therefore ought to improve SpeM's ability to deal with out-of-vocabulary words and with the lexical embedding problem (e.g., through disfavouring the parse *ch apple*, given the input *chapel*). Again, the present simulations show that SpeM can be used to simulate performance in specific psycholinguistic experiments.

### 3.6 General Discussion

In this paper, we attempted to bridge the gap that has existed for decades between the research fields of human and automatic speech recognition. According to Marr (1982), every complex information processing system, including any speech recogniser, can be described at three different levels: the computational, the algorithmic, and the implementational. In the present paper, we offered a computational analysis of speech recognition, with an emphasis on the word recognition process. We focussed initially on the computational level instead of the algorithmic and implementational levels. As we showed, a computational-level description of spoken word recognition applies equally well to computer speech systems as to human listeners, since they both have the same computational problems to solve. The computational-level analysis of the word recognition process revealed close parallels between HSR and ASR. We identified a number of key computational problems that must be solved for speech recognition both by humans and by ASR systems, and we reviewed the standard approaches that have been taken in both HSR and ASR to address these problems.

We illustrated the computational parallels between HSR and ASR by developing SpeM: A model of HSR, based on Shortlist (Norris, 1994), that was built using techniques from ASR. SpeM is not just a reimplementing of Shortlist; it represents an important advance over existing models of HSR in that it is able to recognise words from acoustic speech input at reasonably high levels of accuracy. Our simulations also showed how the representations and processes in SpeM allow it to deal with the computational problems that we highlighted in our review of HSR and ASR. The use of separate prelexical and lexical levels of processing, and, crucially, a probabilistic prelexical level, allows the model to deal quite well with the invariance problem (the problem caused by the variability in the acoustic-phonetic realisation of words in the speech signal). SpeM strongly outperformed Shortlist in its ability to recognise words from spontaneous speech, spoken by a large number of different talkers in a noisy environment, largely due, we showed, to the probabilistic prelexical level in SpeM. We also showed that the combination of a DP lexical search algorithm and phone insertion, deletion, and substitution costs allows SpeM to approach a different aspect of the invariance problem – the fact that real-world

pronunciations of words often diverge, due to the insertion and/or deletion and/or substitution of phonemes, from those words' canonical pronunciations. The probabilistic prelexical level also allows SpeM to recognise speech in close to real time (i.e., it offers a solution to the second computational problem we highlighted, the real-time processing problem).

Our simulations using the input *ship inquiry* showed in addition that SpeM is able to solve the lexical embedding problem (the fact that any stretch of speech is likely to be consistent with several different lexical hypotheses) and the segmentation problem (how can continuous speech be segmented into words when there are no fully reliable cues to word boundaries?). The simulations using the materials from McQueen et al. (1994) and Norris et al. (1997) confirmed that SpeM was able to reproduce their data on lexical competition and the PWC, respectively. In turn, these results also suggest that SpeM is armed to deal with the fifth and final computational problem which we discussed: the out-of-vocabulary problem. Taken together, these simulations illustrate that the theory of HSR underlying SpeM (and Shortlist) holds in the situation of real speech input; in all simulations, the input to SpeM was the acoustic speech signal.

### **3.6.1 Value of SpeM enterprise for HSR**

There are a number of ways in which the comparison of HSR and ASR, and the SpeM model itself, can be of value in advancing our understanding of spoken word recognition in human listeners. The most obvious contribution that ASR can make to theories of HSR is by facilitating development of models that can address the complete range of issues from acoustic analysis to recognition of words in continuous speech. As we have shown with the SpeM simulations reported here, such models can be assessed and evaluated in exactly the same way as existing computational models. One clear advantage of these models is that they can be tested with precisely the same stimulus materials as used in the behavioural studies being simulated, rather than using some idealised form of input representation. These benefits are illustrated by the simulations reported here. First, as in the *ship inquiry* simulations, detailed analyses of hand-crafted (but real speech) inputs can be carried out. Second, as in the lexical competition and PWC simulations, the model can be used to test psycholinguistic theory by comparing its performance on the same set of materials as were presented to listeners in a listening experiment. Analysis of the failures of SpeM informs us about areas where the model needs improvement. As is clear from the present simulations, SpeM's performance is not perfect. We argued in the context of the *ship inquiry* simulations that the limitations of the model given this input were due largely to problems with the APR. These problems are undoubtedly at least part of the reason for the limited success that SpeM had in the other simulations. Obviously, if the APR fails, then everything downstream of the APR must fail too. It may therefore be necessary in HSR modelling to continue to use idealised inputs, in parallel with real-speech simulations in models such as SpeM. Nevertheless, producing a better front end should be one of the

goals of HSR modelling; one challenge for the future will therefore be to establish whether the limitations of SpeM's APR can be overcome.

Of course, producing models that can operate on real speech is not an end in itself. The real benefit of such models is in their contribution to the development of better theories. For example, although HSR modelling has not been naive about the complexity and variability of real speech, it has tended to focus on explaining specific sets of data from experiments (and those experiments have used high quality laboratory speech). HSR modelling has therefore tended to avoid detailed analysis of the problems of robust speech recognition given real speech input. As we noted earlier, the fact that HSR models can not recognise real speech can potentially make it hard to evaluate the theoretical assumptions embodied in those models. It is sometimes difficult to know whether or not a particular theoretical assumption would make a model better or worse at recognising speech, or might even make it fail to recognise speech altogether. ASR modelling has of course been forced to deal with those problems (ASR systems have to be reasonably successful in recognising words in real-world speech communication situations). The ASR approach adopted in SpeM thus offers a new way of looking at specific modelling problems in HSR from the perspective of the technical problem of achieving reasonable levels of recognition of words in real speech.

We have highlighted two areas where we believe that the more formal and practical considerations of building a speech recogniser can inform issues of active theoretical debate in psychology. Although models incorporating process interaction (Section 3.2.6), or episodic recognition (Section 3.2.2) continue to have adherents among psychological researchers, work in ASR throws down a strong challenge to both of these theories: Is it possible to demonstrate any real benefit of on-line interaction, or to show how it might be possible to build a practical large-vocabulary recogniser based on episodic representations?

In addition, the integrated search procedures used in ASR leads to a very different perspective on the interaction debate from that usually adopted in HSR. In the psychological literature the debate is usually seen as a contrast between models with and without interaction between processes responsible for lexical and prelexical processing. The question is: Does lexical information feedback to influence the internal workings of prelexical processes? However, the integrated search processes used in ASR models do not fit neatly into either of these categories. In ASR, there tends not to be independent levels of processing (such as the prelexical and lexical levels). Instead, many different sources of information can contribute to a single lexical search process. Thus, for example, bottom-up acoustic costs can be combined in the search lattice with language model costs that specify the probability of words as a function of syntactic or semantic constraints. In the terminology suggested by Norris (1982) this is an information interaction rather than a process interaction (i.e., it is not the case that, e.g., a syntactic processor influences an acoustic-phonetic processor). Thus, even though the concept of a single, combined search

process may seem alien to psychologists who tend to build models with distinct processing levels, this kind of approach need not involve any process interactions.

Although we have not considered the use of higher level sources of information here, the principle of a unified search process in ASR is usually extended to syntactic and semantic factors too (usually in the form of a ‘language model’). Syntactic or semantic constraints could influence the choice of the best path(s) through the search lattice. This is the most obvious way of dealing with sentence context effects in a model like SpeM; one that is close in spirit to the suggestion (Norris, 1994) that the Shortlist model could be combined with the Checking Model (Norris, 1986) to account for context effects. As we have just argued, however, the inclusion of syntactic constraints as probabilistic biases in the lexical search process would not undermine the assumption that Shortlist and SpeM are non-interactive models. That is, the contextual biases could change the path scores and hence the ultimate segmentation of a given input (i.e., there would be an information interaction) but could not change the bottom-up fit of a word to a stretch of acoustic signal (i.e., there would be no possibility of a process interaction).

The preceding discussion also highlights the fact that the entire word recognition process in both ASR and HSR is best characterised as a search process. The close similarities between the ASR-inspired lattice search process in SpeM and the interactive-activation lexical competition process in Shortlist (see Figure 3-2) make clear that even in a connectionist model with parallel activation of multiple lexical hypotheses, word recognition is a search for the best-matching word(s) for a given input. Put another way, in spite of differences at the algorithmic and implementational levels, word recognition is, computationally, a search problem.

Furthermore, the Bayesian approach adopted in SpeM has implications for many psycholinguistic questions, for instance with respect to the modelling of word frequency effects and with respect to the effects of phonetic mismatch on word recognition. When using Bayes’ Rule to calculate lexical activation, as in SpeM, there is, for example, no need to have an explicit inhibition mechanism to handle mismatching input like the [ʃ] in [ʃIgəɾɛt] (i.e., how a drunk might say the word *cigarette*). The issue in a Bayesian model becomes one of whether  $P(ʃ|s)$  is high, rather than in standard HSR models, where the question, in deriving some mismatch penalty, is whether [s] is confusable with [ʃ]. That is, the Bayesian approach changes one’s way of thinking about spoken word recognition from the notion of *what is similar* to the notion of *what is likely*. Norris et al. (in preparation) also adopt a Bayesian approach in related work developing the Shortlist model. Although the model of Norris et al. uses Bayesian measures, computed on the basis of probabilistic path scores, it differs from SpeM in that it uses input derived from data on perceptual confusions rather than real speech input. That is, rather than using an ASR front end, Norris et al. drive their model from input designed to reflect the characteristics of human

prelexical processing. That paper discusses the theoretical implications of a Bayesian approach to HSR in more detail than is possible here.

In assessing the value of models like SpeM in evaluating theories of HSR, it is worth considering one other point. Some psychologists might be concerned that these ASR techniques do not have the familiar comforting look and feel of, for example, the connectionist models commonly used in psychology. That is, at first glance, connectionist models might seem to be neurobiologically more plausible. However, the contrast between a connectionist model and say, an HMM or Viterbi search, may be nothing more than a difference at the implementational level. We know from Hornik et al. (1989) that connectionist networks are universal approximators. That is, algorithms like Viterbi search could be implemented as connectionist networks. If our analysis is correct, given that the human brain can recognise speech, it must implement algorithms that can compute the appropriate functions. Connectionist networks could only stake a claim to superiority if they could be shown to implement algorithms that could perform the computations necessary for speech recognition, but that could not be implemented in non-connectionist models. For more general arguments in favour of explanations at a computational level, rather than in terms of mechanisms or implementations, the reader is referred to Anderson (1990).

### **3.6.2 Value of SpeM enterprise for ASR**

The SpeM enterprise also has implications for ASR. Most mainstream ASR systems use some kind of integrated search algorithm: they compute the best path through the complete lattice, and then trace back to identify the words that make up that path (see, e.g., Juang & Furui, 2000). SpeM, however, is capable of giving a ranked list of the most likely words at each point in time (i.e., at each node in the input lattice). For each word and each path, SpeM computes an activation value: as long as a word is consistent with the acoustic input, its activation grows. Scharenborg et al. (2003c) show that this feature of SpeM allows the model to recognise words before their acoustic offset. Continuous and early recognition measures could be of considerable value in ASR systems, which often do not provide on-line recognition measures.

Second, there are important lessons to be learned from SpeM for the development of more dynamic ASR systems. As we argued in Section 3.2, both human and machine word recognisers need to be able to adjust their operation to achieve good large-vocabulary speaker-independent recognition performance. We suggested that the prelexical level in SpeM allows for retuning processes that would allow for adjustments to generalise across both speakers and words. Two-stage ASR systems, similar to the cascaded processing described in Section 3.2.2, may, therefore, prove to have more flexibility than traditional one-stage systems. Two-stage procedures have another advantage over one-stage procedures. Because of the intermediate symbolic representation of the speech signal in a two-step recognition system, the second recognition step can be used for integrating more

powerful language models (e.g., morphological, morpho-phonological, morpho-syntactic, and domain knowledge) into the system (see, e.g., Demuyne et al., 2003).

A final implication for ASR also concerns the two-stage architecture of SpeM, and its potential for dynamic adjustment. Many spontaneous speech effects, such as hesitations and repetitions, and the occurrence of out-of-vocabulary words, are problematic for the word-based integrated search in ASR, since this type of search by default tries to match the results of these spontaneous speech phenomena onto lexical items. ASR systems require acoustic garbage models to handle these phenomena. In SpeM, the use of the garbage symbol [?] makes it possible to model speech that does not consist entirely of lexical items. The garbage symbol simply matches with a phone (sequence) that does not match with a lexical item. The combination of the Possible Word Constraint implementation and the garbage symbol makes it possible in SpeM for out-of-vocabulary words to be marked as new words. A garbage symbol sequence that is matched against a sequence of phones containing a vowel can be considered to be a possible word, and could on the basis of this PWC evaluation be added to the lexicon. In this way, new words could be learned, and thus the number of out-of-vocabulary words could be reduced. Although the step of adding new words to the lexicon is not implemented in SpeM, it nevertheless ought to be possible to include similar mechanisms in new and more dynamic ASR systems, in the continued search to improve recognition performance.

### **3.6.3 Limitations of the computational analysis**

As we set out in the introduction, the computational analysis presented here has been restricted to the problem of spoken word recognition. In fact, the scope of our analysis has been restricted to only part of the word recognition problem. We have only touched briefly on questions about the nature of prelexical representations, or the kind of acoustic-phonetic analyses that must form the front-end of a speech recogniser. In part this reflects a conscious decision to focus our discussion on issues where there are clear parallels between ASR and HSR. It also reflects the limitations of our computational analysis, however. When dealing with questions such as lexical competition, there is a clear case to be made that deriving an optimum lexical parse of the input is a central part of the computational task of a speech recogniser. We can also suggest a number of algorithms that might compute the necessary functions. However, as yet we can offer no comparable analysis of the necessary computations required for the early stages of acoustic-phonetic analysis. We could review ASR techniques for extracting spectral and temporal information from the signal, and we could compare them with models of the human auditory system (e.g., Meddis & Hewitt, 1991; Patterson et al., 1995). However, in neither case can we offer a detailed specification of the kind of computation these stages must perform. That we must leave as a challenge for the future.

### 3.6.4 Conclusion

Despite good intentions, there has been little communication between researchers in the fields of ASR and HSR. As we suggested in the introduction, this failure may stem from a lack of common vocabulary. Research in both areas has tended to concentrate on the question of *how* humans or *how* machines recognise speech, and to approach these questions by focussing on algorithms or implementations. Here, we have presented a computational-level analysis of the task of recognising spoken words that reveals the close parallels between HSR and ASR. For almost every aspect of the computational problem, similar solutions have been proposed in the two fields. Of course, the exact algorithms differ, as does everything about how they are implemented, but both fields have had to solve the same problems. The parallels between the two fields are further emphasised by the implementation of the speech-based model of human speech recognition, SpeM. We hope that the computational analysis can provide a common framework to encourage future communication between the disciplines of ASR and HSR. As we have suggested here, each has a great deal to learn from the other.

### Acknowledgements

Part of this work was carried out while the first author was visiting the Medical Research Council Cognition and Brain Sciences Unit, Cambridge, UK. Parts of this research have been reported at the Eurospeech 2003 Conference, Geneva, Switzerland, and at the IEEE workshop on Automatic Speech Recognition and Understanding, St. Thomas, US Virgin Islands.

The authors would like to thank Lou Boves and Anne Cutler for fruitful discussions about this research, Gies Bouwman for his help in implementing SpeM, Diana Binnenpoorte for her help with processing the recordings for the *ship inquiry*, the lexical competition, and PWC simulations, and Gareth Gaskell and two anonymous reviewers for their comments on an earlier version of this manuscript.

## Appendix A

This table displays the 10-best segmentations as calculated by SpeM for the three recordings of *ship inquiry*. The column indicated by ‘Segmentation’ shows the sequence of recognised words in DISC-format (Burnage, 1990). The column ‘Total cost’ shows the total path cost as calculated by SpeM (see Section 3.3.3). The ordering of the paths is done on the basis of the total path cost.

	<b>Recording 1</b>		<b>Recording 2</b>		<b>Recording 3</b>	
	<i>Segmentation</i>	<i>Total cost</i>	<i>Segmentation</i>	<i>Total cost</i>	<i>Segmentation</i>	<i>Total cost</i>
<b>1</b>	SUk INkw2@ri	863.890	JIp INkw2@ri	744.930	SIp INkw2@ri	779.770
<b>2</b>	SUk Ink2@rIN	864.000	SIp INkw2@ri	745.320	JIp INkw2@ri	780.200
<b>3</b>	SIp INkw2@ri	864.080	JIt INkw2@ri	745.400	SIIt INkw2@ri	780.240
<b>4</b>	Sut INkw2@ri	864.180	JVb INkw2@ri	745.660	SUk INkw2@ri	780.330
<b>5</b>	SIp Ink2@rIN	864.190	JIk INkw2@ri	745.700	Sut INkw2@ri	780.370
<b>6</b>	JIp INkw2@ri	864.240	SIIt INkw2@ri	745.780	S@d INkw2@ri	780.560
<b>7</b>	SIIt INkw2@ri	864.280	SUk INkw2@ri	745.870	JIt INkw2@ri	780.670
<b>8</b>	Sut Ink2@rIN	864.290	Sut INkw2@ri	745.920	SIpIN kwQri	780.810
<b>9</b>	JIp Ink2@rIN	864.350	S@d INkw2@ri	745.950	SIbin kwQri	781.080
<b>10</b>	SIIt Ink2@rIN	864.390	JIp INkw2@ri	746.150	SIp Ink2@ rIt	781.130





## ‘Early recognition’ of polysyllabic words in continuous speech

Reformatted from:

O. Scharenborg, L. ten Bosch, and L. Boves. “‘Early recognition’ of polysyllabic words in continuous speech,” *Resubmitted to Computer Speech and Language*.

*Humans are often able to recognise a word before its acoustic realisation is complete. This in contrast to conventional automatic speech recognition (ASR) systems, which compute the likelihood of a number of hypothesised word sequences, and identify the words that were recognised on the basis of a trace back of the hypothesis with the highest eventual score, in order to maximise efficiency and performance. In the present paper, we present an ASR system, SpeM, based on principles known from the field of human word recognition that is able to model the human capability of ‘early recognition’ by computing word activation scores during the speech recognition process.*

*Experiments on 1,463 polysyllabic ‘focus’ words in 885 utterances showed that 64.0% (936) of the focus words were recognised correctly at the end of the utterance. For 81.1% of the 936 correctly recognised focus words the local word activation allowed us to identify the word before its last phone was available, and 64.1% of those words were already identified one phone after their lexical uniqueness point.*

*We investigated two types of predictors for deciding whether a word is considered as recognised before the end of its acoustic realisation. The first type is related to the absolute and relative values of the word activation,  $Act_{min}$  and  $\theta$ , which trade false accepts for false rejects. The second type of predictor is related to the number of phones of the word that have already been processed and the number of phones that remain until the end of the word. The results showed that SpeM’s performance increases if the amount of acoustic evidence in support of a word increases and the risk of future mismatches decreases.*

**Keywords:** *automatic speech recognition; human speech recognition; early speech recognition; continuous speech recognition*

## 4.1 Introduction

For almost all tasks and under almost all conditions humans do a much better job at recognising speech than the most advanced automatic speech recognition (ASR) systems (Lippmann, 1997). Thus, it is not surprising that there are numerous indications that humans employ different algorithms for the processes that are necessary to convert continuous acoustic signals into discrete lexical representations than today’s ASR systems. And it is only natural that several ASR researchers have suggested to take a fresh look at the way humans recognise speech, and try to incorporate the processes that are most likely to make the difference into the design of ASR systems (e.g., Carpenter, 1999; Hermansky, 2001; Moore, 2003).

Most theories of human speech recognition (HSR; Gaskell and Marslen-Wilson, 1997; Luce et al., 2000; McClelland and Elman, 1986; Norris, 1994) assume that human listeners first map the incoming acoustic signal onto prelexical representations (e.g., in the form of phonemes or features) and that these resulting discrete symbolic representations are then matched against the words in an internal lexicon. In general terms, this is not unlike the

way ASR systems operate, although most mainstream systems avoid an explicit representation of the prelexical level to prevent decisions that might incur irrecoverable errors. Looking more closely, however, the lexical search performed by human listeners and ASR systems appears to be organised quite differently. ASR systems postpone final decisions as long as possible (i.e., until additional input data can no longer affect the result). Again, this strategy is chosen in order to prevent premature decisions, the results of which may affect following words. On the other hand, there is ample evidence that human listeners are able to recognise words reliably even before the corresponding acoustic signal is complete (Marslen-Wilson, 1987). According to theories of HSR, human listeners compute a word activation measure (i.e., a measure indicating the extent to which a word is activated based on the incoming speech signal) as the speech comes in and presumably make a decision as soon as the activation of a word is high enough, often before all acoustic information of the word is available (Marslen-Wilson, 1987; Marslen-Wilson and Tyler, 1980; Radeau et al., 2000).

Marslen-Wilson (1987) coined the term *early selection* for the “reliable identification of spoken words, in utterance contexts, *before* sufficient acoustic-phonetic information has become available to allow correct identification on that basis alone.” He reviews a number of gating experiments (a word is being presented in segments of increasing duration, and subjects are asked to identify the word being presented and to give a confidence rating after each segment) and monitoring experiments (detection of a target sequence, which may be embedded in a sentence or list of words/nonwords, or in a single word or nonword) in the context of early selection. On the basis of the results of these experiments, he concluded that in normal speech recognition, content words heard in an utterance context can be selected and recognised earlier than would be possible if just the acoustic input was being taken into account.

Identifying and recognising words before their acoustic realisation is complete is important in human-human communication, for example for adequate turn-taking in a dialogue with minimal response latencies. It may also enhance the segmentation of the continuous stream of acoustic information into words, a process that should be easier if the end of words can be predicted (Marslen-Wilson, 1987). The capability of recognising words on the basis of their initial part certainly helps human listeners in detecting and processing self-corrections, broken words, repeats, etc. (Stolcke et al., 1999).

This paper introduces the concept of ‘*early recognition*’, i.e., the reliable identification of spoken words *before* the end of its acoustic realisation, but *after* the uniqueness point (UP) of the word (given the lexicon). The restriction to recognition at or after the uniqueness point allows us to focus on acoustic recognition, with only a small impact of a language model, which would be comparable – but certainly not identical – to the contexts used in human word recognition in Marslen-Wilson’s definition of ‘early selection’.

If one wants to model early recognition in ASR after human speech recognition, one needs to develop an ASR system that is able to produce a measure analogous to the word activation measure – as used by human listeners – that can be computed on-line, as additional speech comes in. In Scharenborg et al. (2003a, 2003b, accepted), we have presented an end-to-end speech recognition system called SpeM (SPeEch based Model of human speech recognition) that is indeed capable of providing ‘word activations’ that are derived from the log-likelihood values in conventional ASR systems. Since the procedure that converts log-likelihoods into word activations is based on Bayes’ Rule, we use the term ‘Bayesian activation’ along with the more general term ‘word activation’ (Section 4.3). The SpeM system consists of three modules: The first converts the speech signal into a phone graph; the second parses the graph to detect (sequences of) words; the third makes decisions about the recognition of words as more acoustic evidence comes in (Section 4.2). Furthermore, during the lexical search, SpeM provides a list of the most likely path hypotheses at every phone node in the phone graph. This enables SpeM to recognise and accept words *before* the end of an utterance or phrase.

In previous papers (Scharenborg et al., 2003b, accepted), we investigated the performance of SpeM as a standard speech recognition system, which makes decisions about the identity of the words (spoken mainly in isolation) it has recognised after the complete signal has been processed. In this paper, we extend this research by investigating SpeM’s capability for early recognition of spoken words. For standard speech recognition, it suffices to search for the best-scoring path through the search space spanned by the language model, the lexicon, and the acoustic input. In early recognition, on the other hand, an additional decision procedure is needed for accepting a word as being recognised if its local word activation fulfils one or more criteria (Section 4.6).

Early recognition is dependent on the structure and the contents of the lexicon. If a lexicon contains many words that only differ in the last one or two phones, early recognition (on the basis of acoustic input) is more difficult than when the lexicon mainly consists of words which contain many different phone sequences after the lexical uniqueness point. At the same time, it is evident that making decisions on the basis of only a few phones at the beginning of a long word is more dangerous than deciding on the basis of a longer string of word-initial phones. Therefore, we will investigate the impact of the number of phones before and after the UP on the decision criteria that must be applied to the Bayesian activation in early recognition. This should allow us to draw conclusions about the feasibility of early recognition in an ASR system (Section 4.6).

Section 4.2 of this paper introduces SpeM, while Section 4.3 explains the way in which SpeM computes the ‘word activation’ measure in some detail. Section 4.4 briefly describes the speech material used in this study. In order to be able to put the results of SpeM on the task of early recognition into perspective, it is necessary to know how well SpeM performs as a standard ASR system. This issue is taken up in Section 4.5. In that section, we also

define the crucial concept of the ‘Recognition Point’ (RP) of a word, and we analyse the location of the RP in the focus words that were recognised correctly. Finally, Section 4.7 provides a general discussion of the results of this study.

## 4.2 The recognition system

SpeM was developed to serve as an experimental ASR system and at the same time also as a tool for research in the field of HSR. In fact, it is a new and extended implementation of the theory underlying Shortlist, the computational model of human word recognition developed by Norris (1994). Unlike Shortlist and most other computational models of HSR, which take handcrafted symbolic phoneme-like representations of the speech signal as input, SpeM starts from the actual acoustic signal.

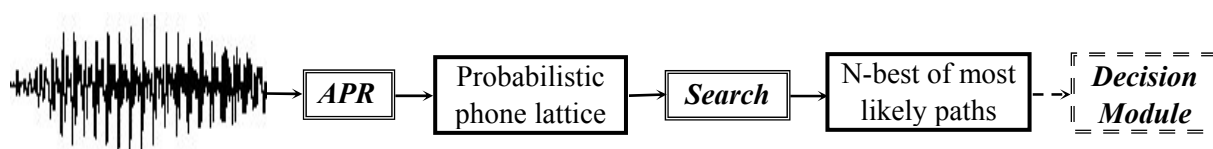


Figure 4-1. Overview of the SpeM model and the additional decision module<sup>1</sup>.

SpeM consists of three modules that operate in sequence (see Figure 4-1). The first module, the automatic phone recogniser (APR), generates a symbolic representation of the speech signal in the form of a (probabilistic) phone graph (Section 4.2.1). The second module, the word search module, parses the graph to find the most likely (sequence of) words, and computes for each word its activation based on, among others, the accumulated acoustic evidence for that word (Section 4.2.2). Below, we give the relevant details of the first two modules. The focus of this paper is on the third module of the system (see Figure 4-1), which makes decisions about the recognition of words as more acoustic evidence comes in. This module is explained in detail in Section 4.6.

The sequential operation of the first two modules should be considered as an implementation detail. It would be easy to change the phone-based architecture of SpeM in such a way that the search module would advance one step each time the APR adds a new node to the phone graph. The essential difference with ASR is that the search module in SpeM depends in a crucial manner on the availability of some kind of prelexical symbolic representation of the speech signal. Consequently, it is not straightforward to implement

<sup>1</sup> In the current research, we are interested in the early recognition of words and not in comparing word activation scores over time in the context of simulating human speech recognition. These two issues require different modules after the search. Therefore, in Figure 4-1, the ‘Evaluation’ module of Figure 3-3 is replaced by the ‘Decision’ module.

early recognition in SpeM in conventional frame-based ASR systems, since in those systems a prelexical symbolic representation is lacking.

### 4.2.1 The automatic phone recogniser

The APR is based on the Phicos ASR system (Steinbiss et al., 1993), but it is easy to build an equivalent module using open source software, such as HTK (Young et al., 2002). For the experiments reported in this paper, 37 context-independent phone models, one noise model, and one silence model were trained on 25,104 utterances in Dutch (81,090 words, corresponding to 8.9 hours of speech excluding leading, utterance internal, and trailing silent portions of the recordings) selected from the VIOS database that consists of telephone calls recorded with the Dutch public transport information system OVIS (Strik et al., 1997). More details about the VIOS database are given in Section 4.4. All phone models and the noise model have a linear left-to-right topology with three pairs of two identical states, one of which can be skipped. For the silence model, a single-state hidden Markov Model (HMM) is used. Each state comprised a mixture of maximally 32 Gaussian densities. The phone models were trained using a transcription generated by a straightforward look-up of the phonemic transcriptions of the words in a lexicon of 1,415 entries, including entries for background noise and filled pauses. For each word, the lexicon contains a single unique phonemic representation, corresponding to the canonical (citation) pronunciation. Pronunciation variation is not taken into account.

The ‘lexicon’ used for the phone recognition by the APR consists of 37 Dutch phones and one entry for background noise, yielding 38 entries in total (in the lexicon, no explicit entry for silence is needed). During recognition, the APR uses a bigram phonotactic model trained on the canonical phonemic transcriptions of the training material.

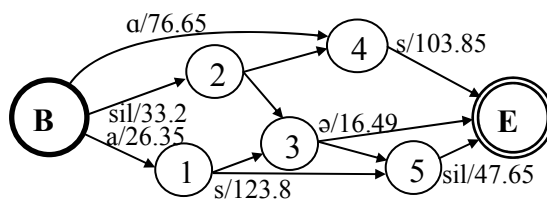


Figure 4-2. A graphical representation of a weighted probabilistic input phone lattice. For the sake of clarity, not all phones and acoustic costs are shown.

The APR converts the acoustic signal into a weighted probabilistic phone lattice without using lexical knowledge. Figure 4-2 shows a simplified weighted phone lattice: The lattice has one root node (‘B’) and one end node (‘E’). Each edge (i.e., connection between two nodes) carries a phone and its bottom-up evidence in terms of negative log likelihood (its acoustic cost). The acoustic cost denotes the probability that the acoustic signal  $X$  was

produced given the phone ( $P(X|Ph)$ , in which  $Ph$  denotes a phone). In our experiments, the acoustic scores for a phone typically range from 10 to 120 (not normalised for length).

#### 4.2.2 The search module

In the lexical search module, the search for the best-matching sequence of words is in effect the search for the cheapest path through the product graph of the input phone lattice and a lexicon represented as a lexical tree. In the lexical tree, entries share common phone prefixes (called word-initial cohorts), and each complete path through the tree represents a pronunciation of a word. See Figure 4-3 for a graphical representation of the beginning of a lexical tree. The lexical tree has one root node (‘B’) and as many end nodes as there are words in the lexicon. The hash ‘#’ indicates the end of a word; the phonemic transcription in the box is the phonemic representation of the complete word. Each node in the lexical tree represents a word-initial cohort. The phonemic transcriptions belonging to the word-initial cohorts are not explicitly shown. Note that the word [as] is an example of an embedded word, since the node labelled with [as] in the lexical tree (Figure 4-3, node 2) has outgoing arcs (thus in this case the phonemic transcription [as] also represents a word-initial cohort). Finally, SpeM supports the use of unigram and bigram language models, which models the prior probability of observing a word and of observing a word given its predecessor. In the experiments reported in this paper, only a unigram language model is used (see also Section 4.4).

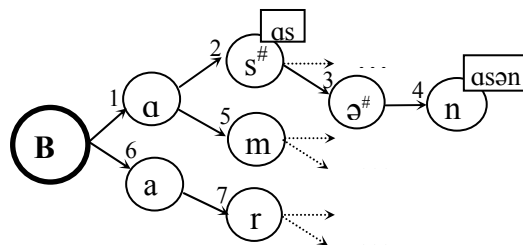


Figure 4-3. A graphical representation of the beginning of a lexical tree.

The search is implemented using dynamic programming (DP) techniques, and is time-synchronous and breadth-first. SpeM calculates scores for each path (the *total cost*), and also a score for the individual words on a path (the *word cost*). The total cost of a path is defined as the accumulation along the path arcs of the bottom-up acoustic cost (as calculated by the APR) and a number of costs associated with SpeM’s parameters. SpeM has a number of parameters that can be tuned individually and in combination. Most of these parameters (e.g., a word entrance penalty and the trade-off between the weights of the bottom-up acoustic cost of the phones and the contribution of the language model) are similar to the parameters in conventional ASR systems. In addition, however, SpeM has two types of parameters that are not usually present in conventional ASR systems. The first novel parameter type is associated to the cost for a symbolic mismatch between the input

lattice and the lexical tree due to phone insertions, deletions, and substitutions. Insertions, deletions, and substitutions have their own weight that can be tuned individually. Because the lexical search in SpeM is phone based, mismatches can arise between the phonemic representation of the input in the phone graph and the phonemic transcriptions in the lexical tree. It is therefore necessary to include a mechanism which explicitly adjusts for phone-level insertions, deletions, and substitutions. In mainstream ASR, however, the search space is usually spanned effectively by the combination of the pronunciation variants in the system's dictionary and the system's language model, so that explicit modelling of insertions, deletions, and substitutions on the phone-level is not necessary.

The second novel parameter type is associated to the *Possible Word Constraint* (PWC, Norris et al., 1997). The PWC determines whether a (sequence of) phone(s) that cannot be parsed as a word (i.e., a lexical item) is phonotactically well formed (being a possible word) or not (see also Scharenborg et al., 2003b, accepted). In SpeM, the PWC is implemented using 'garbage' symbols, comparable to the 'acoustic garbage' models in ASR systems. The garbage symbol in SpeM matches all phones with the same cost (note that the acoustic costs of the phones themselves do vary) and is hypothesised whenever an insertion that is not word-internal occurs on a path. A garbage symbol (or an uninterrupted sequence of garbage symbols) is itself regarded as a word, so the word entrance penalty is added to the total cost of the path when garbage appears on that path. The PWC evaluation is applied only to paths on which garbage is hypothesised. Word onsets and offsets, plus utterance onsets and offsets and pauses, count as locations relative to which the viability of each garbage symbol (or sequence of symbols) is evaluated. If there is no vowel in the garbage sequence between any of these locations and a word edge, the parse is penalised and the PWC cost is added to the total cost of the path. For example, consider the utterance "they met a fourth time", where the last sound of the word *fourth* is pronounced as [f]. If *fourf* is not stored as a possible pronunciation in the lexicon, a potential parse by the recogniser in terms of lexical items is *they metaphor f time*. Since the phone 'f' is not a possible word in English, the PWC mechanism penalises this parse, and if the cost of the substitution of [θ] by [f] is less than the PWC cost, the parse yielding the word sequence '*fourth time*' will win. At the same time, it is worth mentioning that the presence of the garbage phones enables SpeM to parse input with broken words and disfluencies, since it provides a mechanism for handling arbitrary phone input (for more information, Scharenborg et al., accepted).

All parameters in SpeM are robust: Even if they are not optimised in combination, SpeM's output does not change significantly if the value of the parameter that was optimised with fixed values of other parameters is changed within reasonable bounds. In this study, the parameters were tuned on an independent tuning set (see Section 4.4), and subsequently used for processing the test corpus.



The output of SpeM consists of an  $N$ -best list of hypothesised parses. Each parse consists of words, word-initial cohorts, garbage, silence, and any combination of these, except that a word-initial cohort can only occur as the last element in the parse. Thus, in addition to recognising full words, SpeM is able to recognise partial words. Furthermore, for each recognised item, its activation and the activation of the entire path up to that point in time are calculated. This capability allows SpeM to simulate results from psycholinguistic experiments on word recognition which show how words are activated over time; it also enables SpeM to provide the activation values that can be used in the decision module where early recognition is decided upon.

### 4.3 The computation of word activation

The functionality of SpeM that is most important here is the computation of word activation. The measure of *word activation* in SpeM was originally designed to simulate experimental results of human word recognition experiments (Scharenborg et al., 2003a, accepted). In the computation of the word activation, the local negative log-likelihood scores for paths and words on a path are converted into activation scores that obey the following properties:

- The word that matches the input best, thus having the smallest *word cost* (see Section 4.2.2), must have the highest activation.
- The activation of a word that matches the input must increase each time an input phone is processed.
- The measure must be appropriately normalised. That is, word activation should be a measure that is meaningful, both for comparing competing word candidates, and for comparing words at different moments in time.

The way SpeM computes word activation is based on the idea that word activation is a measure related to the bottom-up evidence of a word given the acoustic signal: If there is evidence for the word in the acoustic signal, the word should be activated. Activation should also be sensitive to the prior probability of a word – even if this effect was not modelled in the original version of Shortlist (Norris, 1994). This means that the word activation of a word  $W$  is closely related to the probability  $P(W|X)$  of observing a word  $W$ , given the signal  $X$ , the cost function maximised in virtually all ASR systems. Thus, it is reasonable to stipulate that the word activation  $Act(W|X)$  is a function of  $P(W|X)$ , and apply the same Bayesian formulae that form the basis of virtually all theories in ASR to estimate  $P(W|X)$ . This is why we refer to  $Act(W|X)$  as the ‘Bayesian activation’. It is important to emphasise that the theory underlying word activation does not require that the sum of the activations of all active words should add to some constant (e.g., 1.0, as in probability theory). In accordance with conventional ASR systems, for the purpose of early recognition it is not mandatory that the total ‘activation mass’ is normalised, as long as it is possible to apply (possibly context dependent) decision thresholds to the measure.

Following Bayes' Rule, we define the word activation  $Act(W|X) = P(W|X)$ , which can be written as:

$$Act(W | X) = \frac{P(X | W)P(W)}{P(X)}, \quad (4-1)$$

Since we also want to deal with incompletely processed acoustic input (for early recognition of words), Equation 4-1 is extended to:

$$Act(W(n) | X(t)) = \frac{P(X(t) | W(n))P(W(n))}{P(X(t))}, \quad (4-2)$$

where  $W(n)$  denotes a phone sequence of length  $n$ , corresponding to the *word-initial cohort* of  $n$  phones of  $W$ . Note that  $n$  is discrete because of the segmental representation of the speech signal.  $X(t)$  is the gated signal  $X$  from the start of  $W(n)$  until time  $t$  (corresponding to the end of the last phone included in  $W(n)$ ).  $P(X(t))$  denotes the prior probability of observing the gated signal  $X(t)$ .  $P(W(n))$  denotes the prior probability of  $W(n)$ .  $W(5)$  may, for example, be /amstə/, i.e., the word-initial cohort of the word 'amsterdam'. In the experiments reported in this paper,  $P(W(n))$  is exclusively based on the unigram probability of the word-initial cohorts and the words.

The (unnormalised) conditional probability  $P(X(t)|W(n))$  in Equation 4-2, is calculated by SpeM as:

$$P(X(t) | W(n)) = e^{-a \cdot TC}, \quad (4-3)$$

where  $TC$  is the total bottom-up cost associated with the word starting from the beginning of the word up to the node corresponding to instant  $t$ .  $TC$  includes not only the acoustic costs in the phone lattice, but also the costs contributed by substitution, deletion, and insertion of symbols (like the acoustic cost calculated by the APR,  $TC$  is a negative log likelihood score). The definition of the total bottom-up cost is such that  $TC > 0$ . The value of  $a$  determines the contribution of the bottom-up acoustic scores to the eventual activation values. The  $a$  weights the relative contribution of  $TC$  to  $Act(W(n)|X(t))$ , so it acts similar to the language model factor in standard ASR systems. To illustrate the effect of  $a$ , consider a cohort  $W(n)_1$  on one path and a different cohort  $W(n)_2$  on a competing path with the same

history as  $W(n)_1$  ( $P(X(t)|\text{history})$  is identical) and an identical LM score ( $P(W_1(n)) = P(W_2(n))$ ). The difference in word activation between  $W(n)_1$  and  $W(n)_2$  is now completely determined by the difference in acoustic scores  $P(X(t)|W_1(n))$  and  $P(X(t)|W_2(n))$  between the two words.  $a$  is a positive number; its numerical value is determined such that the three properties of word activation introduced at the start of this section will hold. The comparison of the results of HSR experiments and SpeM simulations (Scharenborg et al., 2003a), yielded a value  $a=0.01$ . In the phone graphs generated of our test material by the APR, the average acoustic score (in terms of negative log likelihoods) of a matching phone is 25. In combination with  $a=0.01$ , this amounts to  $P(X(t)|W(n)) \approx \exp(-0.25) \approx 0.78$ , if  $W(n)$  is one phone long (i.e., if  $n=1$ ).

In SpeM, in contrast to conventional ASR systems, the prior  $P(X(t))$  in the denominator of Equation 4-2 cannot be discarded, because hypotheses covering different numbers of input phones must be compared. The problem of normalisation across different paths is also relevant in other unconventional ASR systems (e.g., Glass, 2003). The denominator, then, is approximated by

$$P(X(t)) = D^{\#nodes(t)}, \quad (4-4)$$

where  $D$  is a constant ( $0 < D < 1$ ) and  $\#nodes(t)$  denotes the number of nodes in the cheapest path from the beginning of the word up to the node associated with  $t$  in the input phone graph. In combination with  $a$ ,  $D$  plays an important role in the behaviour over time of  $Act(W(n)|X(t))$ . Once the value of  $a$  is fixed, the value of  $D$  follows from two constraints: 1) the activation on a matching path should increase; 2) the activation on any mismatching path should decrease. Then it follows from Equation 4-2 and these two additional requirements that:

$$e^{-a(\text{avgMismatchPhone} + \text{SubC})} \leq D \leq e^{-a(\text{avgMatchPhone})}, \quad (4-5)$$

where  $\text{avgMismatchPhone}$  is the average acoustic cost of a mismatching phone on a competing path,  $\text{SubC}$  is the cost for a phone substitution, and  $\text{avgMatchPhone}$  is the average acoustic cost of a matching phone on the first-best path. Because of the way the APR works, the average acoustic cost of a mismatching phone is only marginally smaller than the average acoustic cost of a matching phone. Thus, the difference between  $(\text{avgMismatchPhone} + \text{SubC})$  and  $\text{avgMatchPhone}$  is essentially determined by the value of  $\text{SubC}$ . The tuning experiments to be described in Section 4.4 yielded  $\text{SubC} = 150$ . The

left-most term in Equation 4-5, then, evaluates to  $\exp(-0.01(26+150)) \approx 0.17$ ; the rightmost term evaluates to  $\exp(-0.01 \cdot 25) \approx 0.78$ . We set  $D = 0.7$ .

Our choice to normalise the Bayesian activation by the expression given by Equation 4-4 is based on two considerations. Firstly, given the Bayesian paradigm, it seems attractive to use a measure with the property that logarithmic scores are additive along paths. Let  $X_1$  and  $X_2$  be two stretches of speech such that  $X_2$  starts where  $X_1$  ends, associated with two paths  $P_1$  and  $P_2$  in the phone lattice (such that  $P_2$  starts where  $P_1$  ends), then  $\log(P(X_1)) + \log(P(X_2)) = \log(P(X_1 : X_2))$  (where ‘:’ means ‘followed by’). This means that the lengths of  $X_1$  and  $X_2$  are assumed to be independent, which is a plausible assumption. Secondly, the normalisation as given by Equation 4-4 is similar to the normalisation that has to be performed in the calculation of confidence measures. In order to be able to compare confidence measures of hypotheses with unequal length, the normalisation must, in some way, take into account the duration of the hypotheses. Equation 4-4 can be regarded as a normalisation in which the number of phones is the normalising factor, rather than the number of frames, that is, as a type of normalisation that is more phonetically oriented.

#### 4.4 Material

There is a considerable phonological overlap among words, because of which any given word is likely to begin and end in the same way as several other words (Luce, 1986). In addition, longer words are likely to have shorter words embedded within them (McQueen et al., 1995). Consequently, short words are likely to have a UP that is not before the end of the word, making it impossible to recognise the word before its acoustic offset. Furthermore, Grosjean (1985) pointed out that especially function words and short infrequent content words may not even be identified by human listeners until the word following it has been heard. Therefore, in our evaluation of SpeM’s ability for early recognition, we focus on polysyllabic content words.

The VIOS training and test corpus consists of utterances taken from dialogs between customers and an automatic timetable information system (Strik et al., 1997). We decided to define a set of 318 polysyllabic station names as *focus* words. From the VIOS database, 1,106 utterances (disjoint from the corpus used for training the acoustic phone models) were selected to tune and test SpeM. Each utterance contained two to five words, at least one of which was a focus word (708 utterances contained multiple focus words). 885 utterances of this set (80% of the 1,106 utterances) were randomly selected and used as the independent test corpus. The total number of focus words in the test corpus was 1,463; 563 utterances contained multiple focus words. The remaining 221 utterances were used as development test set and served to tune the parameters of SpeM (see also Section 4.2.2). The parameter settings yielding the lowest Word Error Rate (WER) were used for the experiment. The WER is defined as:

$$WER = \frac{\#insertions + \#deletions + \#substitutions}{N} \cdot 100\%, \quad (4-6)$$

The insertions, deletions, and substitutions in Equation 4-6 concern words (different from the phone insertions, deletions, and substitutions discussed in the previous sections);  $N$  is the number of words in the reference transcription.

The lexicon used by SpeM in the test consisted of 980 entries: The 318 polysyllabic station names, additional city names, verbs, numbers, and function words. There are no out-of-vocabulary words. For each word in the lexicon, one unique canonical phonemic representation was available. A unigram language model (LM) was trained on the VIOS training data – the same data that was used for training the acoustic models and the bigram phonotactic model for the APR.

#### 4.5 Early recognition

In this section, we first present the results of an experiment designed to get an idea of how SpeM performs as a standard ASR system (Section 4.5.1). The results are presented in terms of WER (for all words in the test set, thus not only the focus words) by taking the best matching sequence of words as calculated by SpeM after processing the entire input and comparing it with the orthographic transcriptions of the test corpus.

Second, we investigate how many of the focus words have a recognition point that is before the end of the word, and thus can, in principle, be recognised before their acoustic offsets during the recognition process (Section 4.5.2). To that end, we investigate the behaviour of the Bayesian word activation score as a measure to rank path and word hypotheses dynamically.

In the analysis, we first determine the proportion of the focus words that were recognised correctly at the end of the utterance. Subsequently, the *recognition point* (RP) was determined, which is defined as the node after which the activation measure of a correct focus word exceeds the activation of all competitors, and remains higher until the end of the word (after the offset of a word, the word’s activation does not change). This means that a word that is not recognised correctly does not have an RP. The RP is expressed as the position of the corresponding phone in the phonemic (lexical) representation of the word. In our analysis, the RP will be related to both the length of the canonical phonemic representation and the lexical uniqueness point (UP) of the word. Prior to the UP, multiple words (in the word-initial cohort) share the same lexical prefix, and therefore cannot be distinguished on the basis of the acoustic evidence.

### 4.5.1 The performance of SpeM as a standard speech recognition system

To get an idea of the task SpeM is facing, we determined the average depth of the input graphs. For all graphs, the number of arcs was divided by the number of nodes. The sum of these averages was then divided by the total number of phone graphs (885). The average depth of all input graphs was 6.3. Thus, on average, SpeM needs to evaluate 6.3 arcs (or phones) at any point in time. Or in other words, each node in the input graph has on average 6.3 outgoing arcs.

Of the 1,463 focus words, 64.0% (936 focus words) were recognised correctly at the end of the word. An analysis of the phone graphs revealed that 309 utterances (34.9% of the test utterances) did not contain a path that matched exactly with the canonical representation of the spoken words. For 95 of these utterances (30.7%), SpeM was able to correctly recognise the (one or more) focus word(s). Thus, SpeM is able to ‘repair’ part of the deficiencies in the output of the APR.

The WER obtained by SpeM on *all* words in the test material was 40.4%. Thus, the performance on polysyllabic station names is only slightly better than the overall WER. This result is certainly worse than the best performance of other ASR systems on the VIOS database observed in previous experiments (Kessens et al., 1999, 2003; Wester, 2003). However, the performance of SpeM as an ASR system cannot be compared directly to results presented for the VIOS database in previous publications. There are a number of reasons for this. First of all, contrary to SpeM, the ASR systems used in previous experiments used bigram language models, while SpeM only used a unigram language model. Second, the subset of the VIOS test set used in the present study contains the longest utterances, which are most difficult to recognise, while previous results were obtained on the full test set, including a large number of *yes/no* answers that appear to boost performance substantially. Finally, the present model uses a two-step recognition procedure in which the APR generated many phone sequences that do not occur in the canonical representations of the words in the lexicon of SpeM. In contrast, previous results were obtained with an ASR in which the acoustic signal could be directly matched against the lexicon (and therefore avoided considering phone sequences that do not occur in the canonical representations of the words).

In the present study, no attempt has been made to maximise the performance of the acoustic model set of the APR. Quite probably, an APR that computes more accurate acoustic likelihoods should allow SpeM to reach a performance level comparable to a conventional ASR system. The results presented in Scharenborg et al. (2003b, accepted) show that SpeM’s performance is comparable to that of an off-the-shelf ASR system (with an LM in which all words are equally probable) when the acoustic model set used to construct the phone graph is optimised for a specific task.

Despite the mediocre performance of SpeM as an ASR system, and although there is still room for improvement of SpeM’s performance as a standard ASR system, it is possible to use SpeM to investigate early recognition, since there are a sufficiently large number of words recognised correctly.

#### 4.5.2 Recognition point analysis

Of the focus words that were recognised correctly, 81.1% had their RP *before* the end of the word (759 of 936 correctly recognised focus words; 51.9% of all focus words). Not all focus words that were recognised correctly have an RP before the end of the word, since a focus word that is correctly recognised by SpeM does not necessarily have a one hundred percent match with the phone sequence in the phone graph. As indicated before, for 34.9% of the utterances, the canonical phone transcription of the utterance was not present in the phone graph. This implies that for many of the focus words phone insertion, deletion, and substitution penalties are added to the total score of the word and the path. Obviously, multiple words can have a small distance to the path through the phone graph that carries the correct solution. Therefore it is clear that the best matching word can only be determined with certainty after all information of all competing words is available.

For the 936 focus words that were ultimately correctly recognised, the RP was related to the UP and to the total number of phones of the word. The results are shown in the form of two histograms in Figure 4-4. The frequency is given along the y-axis. In the left panel, the x-axis represents the distance (in phones) between the UP and the RP of the focus words.  $N = 0$  means that the word activation exceeded all competitors already at the UP. In the right panel, the x-axis represents the position of the RP (in number of phones ( $N$ )) relative to the last phone in the canonical representation of the word. Here,  $N = 0$  means that the word activation exceeded the competitors only at the last phone of the word.

For the interpretation of the information in Figure 4-4, the phonemic structure of the words in the set of correctly recognised 936 focus words and the position of the UP of the words must be known. This information is shown in Table 4-1. The first column shows the distance in number of phones between the UP and the end of the word. ‘Total-UP’ = 0 means that the UP is at the end of the word: The word is embedded in a longer word. Columns 2 and 3 show the number of focus word types and tokens with ‘Total-UP’ phones between the end of the word and the UP. From Table 4-1 it can be deduced that the UP of 85.0% of all focus word tokens (1,243/1,463) is at least two phones before the end of the word; only 2% of the focus word tokens (30/1,463) have their UP at the end of the word. The high frequency in the case of  $N = 3$  in the right panel of Figure 4-4 is due to an idiosyncratic characteristic of the data. As can be seen in Table 4-1, there is a large set of words that have their UP three phones before the end of the word (450).

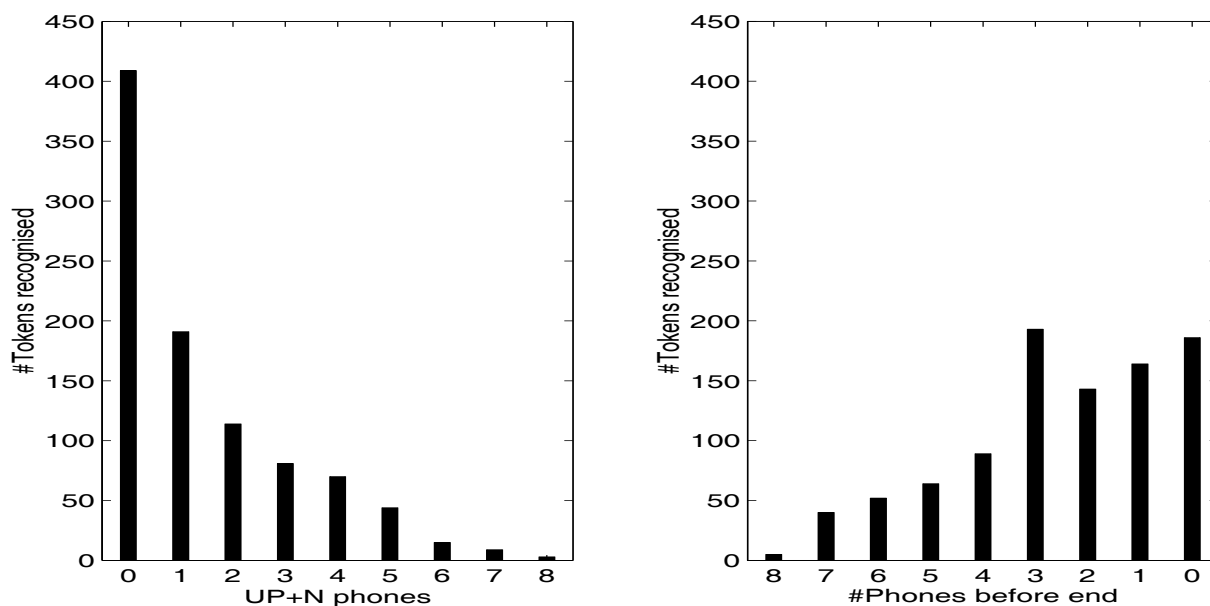


Figure 4-4. In the left panel, recognition point related to the uniqueness point (‘UP+N phones’); in the right panel, the recognition point related to the total number of phones in the word (‘#Phones before end’) for the 936 correctly recognised focus words.

Table 4-1. The distribution (in #types and #tokens) in number of phones between the UP and the total number of phones in the word (‘Total-UP’); Cumulative: #focus word tokens that could in principle be recognised at position Total-UP.

Total-UP	#types	#tokens	Cumulative
9	2	4	4
8	3	11	15
7	17	57	72
6	38	82	154
5	39	186	340
4	50	271	611
3	63	450	1,061
2	50	182	1,243
1	44	190	1,433
0	10	30	1,463



Combining the information in Figure 4-4 and Table 4-1 reveals that although only 2% of the focus words have their UP at the end of the word, 19.8% (185/936, see right panel of Figure 4-4) of the words were only recognised at the end of the word. Apparently, SpeM is not always able to recognise a word before its acoustic offset, despite the fact that the UPs in the set of words were almost always at least one phone before the end of the word. More interestingly, however, from Figure 4-4 it can also be deduced that 64.1% (sum of  $N = 0$  and  $N = 1$ , see left panel of Figure 4-4) of the total number of recognised focus words were already recognised at, or maximally one phone after the UP. Taking into account that 85.0% of the focus words have at least two phones after their UP, this indicates that SpeM is able to take advantage of the redundancy caused by the fact that many words in the vocabulary are unique before they are complete.

#### 4.6 Predictors for reliable on-line early recognition

The experiment presented in the previous section showed that the word activation of many polysyllabic content words exceeds the activation of all competitors already before the end of the words. However, this does not imply that word activation can be safely used to perform early recognition. If we want to use word activation as a basis for deciding whether a word is considered as recognised before the end of its acoustic realisation, we must develop a decision procedure. To that end, we have experimented with a combination of absolute and relative values. In addition, we have investigated whether the reliability of early decisions is affected by the number of phones of the word that have already been processed and the number of phones that remain until the end of the word.

In Section 4.6.1, we explain the decision module that we implemented. The performance of that module will be evaluated in terms of *precision* and *recall*:

*Precision*: The total number of *correctly* recognised focus words relative to the total number of recognised focus words. Precision gives an impression of the trade-off between correctly recognised focus words and false accepts.

*Recall*: The total number of *correctly* recognised focus words divided by the total number of focus words in the input. Recall gives an impression of the trade-off between correctly recognised focus words and false rejects.

As usual, there is a trade-off between precision and recall. Everything else being equal, increasing recall tends to decrease precision, while increasing precision will tend to decrease recall. We are not primarily interested in optimising SpeM for a specific task in which the relative costs of false accepts and false rejects can be established, since we are mainly interested in the feasibility of early recognition in an ASR system. Therefore, in contrast to standard procedure, we decided to refrain from defining a total cost function that combines recall and precision into a single measure that can be optimised.

### 4.6.1 Decision Module

For a focus word to be recognised by SpeM, the following three conditions have to be met:

1. The phone sequence assigned to the focus word is at or beyond the focus word's UP.
2. The *quotient* of the word activation of the focus word on the best-scoring path and the word activation of its closest *competitor* (if present) exceeds a certain threshold ( $\theta$ ). Thus, we do not want SpeM to make a decision as long as promising competitors are still alive. The notion that there must be a sufficiently large difference between the first best hypothesis and its runner-up has also been used for a long time in various types of ASR systems to compute a kind of confidence measure (e.g., Brakensiek et al., 2003). In the SpeM search, two words are said to be in competition if the paths they are on contain an identical sequence of words, except for the word under investigation. Figure 4-5 illustrates this with an example where the first-best path: [a:vɔnt vo:rbYr\*] competes with the path: [a:vɔnt xu:dəm\*]. The competitor of [vo:rbYr\*] is thus [xu:dəm\*]. The asterisk indicates that the processing of a word has not yet reached its last phone.

Given our definition of ‘competitor’ it is not guaranteed that all words always have a competitor, because it is possible that all paths in the  $N$ -best list are completely disjunct – and so do not share the same history, as is required for being competitor. Absence of a competitor makes the computation of  $\theta$  impossible. To prevent losing all words without competitors due to a missing value, we accept all focus words without a competitor that appear at least five times in the  $N$ -best list. In the experiments described below, we tested various values for  $\theta$ . The number of hypotheses in the  $N$ -best list is set at 10, so that SpeM will output the 10 most likely hypotheses for each node in input graph.

3. The value of the Bayesian activation of the focus word itself should exceed a certain *minimum activation* ( $Act_{min}$ ). Thus, SpeM does not just accept the word with the highest activation, irrespective of the absolute value of the activation. In the experiments described below, we tested various values for  $Act_{min}$ .  $Act_{min}$  is reminiscent of the graph based confidence measures introduced in Wessel et al. (2001).

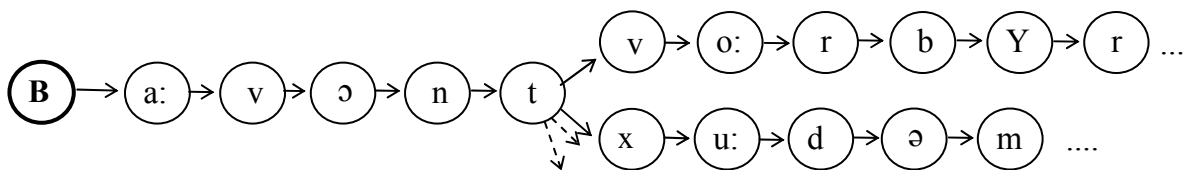


Figure 4-5. Two focus words in competition.

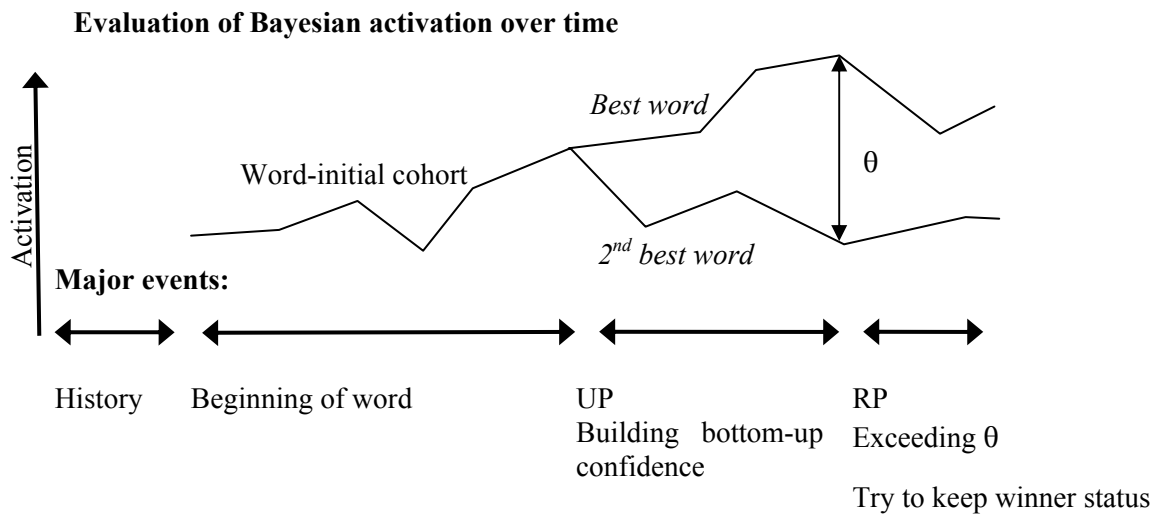


Figure 4-6. Schematic illustration of the process of (on-line) early recognition.

The process of early recognition is schematically depicted in Figure 4-6. The word activation of words grows over time as matching evidence is added. Before the word’s UP, several words are consistent with the phone sequence; the difference in activation of the individual words in the cohort is caused by the influence of the LM. After a word’s UP, it has its own word activation. For the purpose of the experiments in this section, we define the *decision point* (DP) as the point at which a word on the first best path meets the decision criteria described in this section.

#### 4.6.2 $\theta$ and $Act_{min}$ as predictors of on-line early recognition

In this section, we investigate the Bayesian activation as a predictor of early speech recognition as a function of  $\theta$  and  $Act_{min}$ . Figure 4-7 shows the relation between precision (y-axis) and recall (x-axis) for a number of combinations of the two thresholds. The symbols on the lines in Figure 4-7 represent the values of  $Act_{min}$  for three different values of  $\theta$ . The value of  $Act_{min}$  was varied between 0.0 and 2.0 in 20 equal-sized steps. The left-most symbol on each line corresponds to  $Act_{min}=2.0$ ; the right-most one corresponds to  $Act_{min}=0.0$ . For the sake of clarity, Figure 4-7 is limited to three values of  $\theta$ ; all other values of  $\theta$  show the same trend.

The results in Figure 4-7 are according to expectation. Recall should be an inverse function of  $\theta$ : the smaller  $\theta$  becomes, the less it will function as a filter for words that have a sufficiently high activation, but which still have viable competitors. Similarly for  $Act_{min}$ : For higher values of  $Act_{min}$ , fewer focus words will have an activation that exceeds  $Act_{min}$ , and thus fewer words are recognised. These results indicate that the Bayesian activation can be used as a predictor for the early recognition of polysyllabic words.

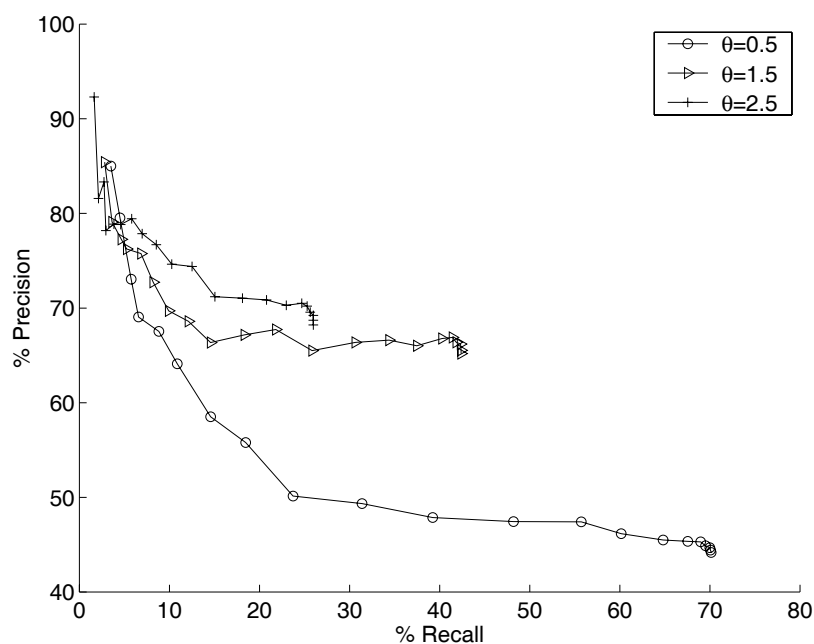


Figure 4-7. For three values of  $\theta$ , precision and recall of all 21 values of  $Act_{min}$  are plotted.

### 4.6.3 The effect of the length of the word

As pointed out before, in our definition of early recognition, a word can only be recognised at or after its UP. Thus, words that have an early UP can fulfil the conditions to be recognised while there is still little evidence for the word. This raises the question whether the amount of evidence in support of a word (the number of phones between the start of the word and the DP), or the ‘risk’ (in the form of the number of phones following the DP until the end of the word) can be helpful in increasing precision and recall. This is the focus of the analyses described in this section. The value for  $Act_{min}$  is set to the arbitrarily chosen value of 0.5; the value of  $\theta$  was varied between 0.0 and 2.0 in 80 equal-sized steps.

We are interested in the number of words that could in principle be recognised correctly at a certain point in time. Therefore, for calculating precision and recall, only the number of focus word tokens that in principle could be recognised correctly should be taken into account. The column ‘Cumulative’ in Table 4-1 shows the number of focus word tokens that could in principle be recognised correctly at ‘Length-UP’ phones before the end of a word. For instance, at 8 phones before the end of the word, the only words that could in principle be recognised correctly are those that have a distance of 8 or more phones between the end of the word and the UP. At 0 phones before the end of a word, all words could in principle be identified correctly. For calculating recall, the total number of correctly recognised focus words is divided by the total number of focus words that could in principle have been recognised correctly. Precision is calculated in the same manner: the

total number of correctly recognised focus words *so far* is divided by the total number of recognised focus words *so far*. The effect of the amount of evidence is investigated in a similar fashion. Precision and recall are computed as a function of the number of phones between the start of the word and the DP, and again, only the number of focus word tokens that in principle could be recognised correctly is taken into account.

The contour plots in Figure 4-8 show the relation between the number of phones separating the DP from the end of the word and precision and recall for different values of  $\theta$ . On the y-axis, the value of  $\theta$  is shown; the x-axis shows the number of phones between the DP and the end of the word. The lines in the plots are the equal-percentage lines for the cumulative precision (upper panel) and the cumulative recall (lower panel). Precision and recall of a point between two equal-percentage lines can be estimated using the distance of the point to the two neighbouring equal-percentage lines. For instance, for  $\theta=1.0$  and a distance of four phones between the DP and the end of the word, precision is about 39%. Figure 4-8 suggests that precision and recall at DPs where there is a high number of phones separating the DP from the end of the word can be rather high (see the bottom left part of Figure 4-8). However, this is an artefact caused by the special characteristics of the 15 focus words that happen to be unique already 8 phones before the end of the word. Precision and recall of distances between 8 and 5 are rather low. However, distances of 4 phones or less show a clear increase in both precision and recall.

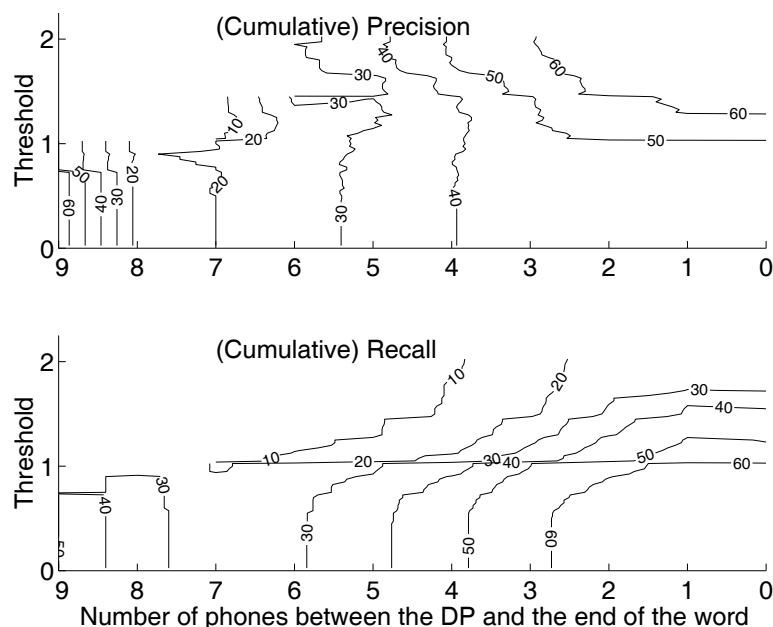
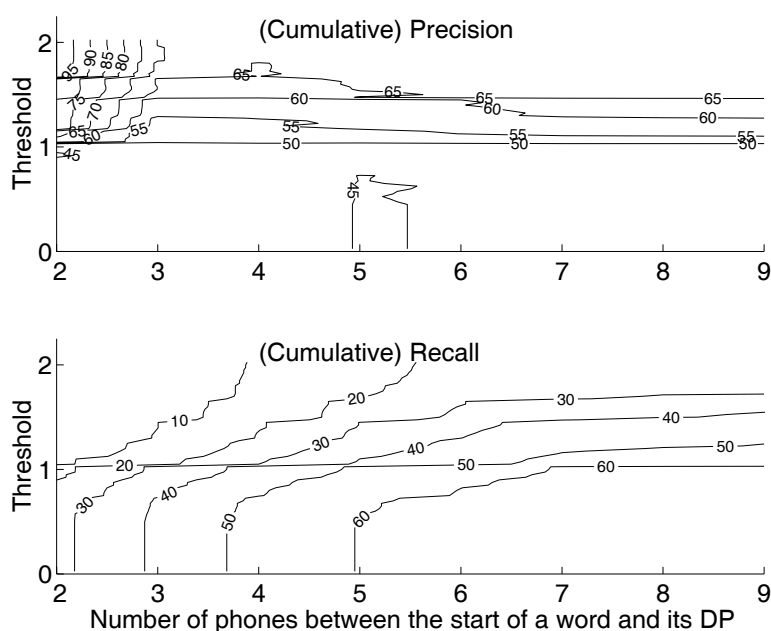


Figure 4-8. The x-axis shows the number of phones between the DP and the end of the word; the y-axis shows the value of  $\theta$ . The upper panel shows precision; the lower shows recall.



*Figure 4-9.* The x-axis shows the number of phones between the start of the word and its DP; the y-axis shows the value of  $\theta$ . The upper panel shows precision; the lower shows recall.

The contour plots in Figure 4-9 show the relation between the number of phones between the start of the word and its DP and precision and recall for different values of  $\theta$ . In other words, Figure 4-9 shows the effect of the amount of information available for a word on precision and recall. The results shown in Figure 4-9 reveal – not surprisingly – that when there is yet little evidence available for the word, recall is rather low. The more phones have been processed, the higher recall is. The high precision for the situation where only two phones have been processed and high values of  $\theta$  is an artefact of the data (see top left part of Figure 4-9) – there are only a few words that exceed the threshold  $\theta$ .

To clarify the effects of an increasing number of phones between the DP and the end of the word and an increasing number of phones between the start of the word and its DP, the precision and recall are plotted for a single  $\theta$  and  $Act_{min}$  value, viz.  $\theta=1.625$  and  $Act_{min}=0.5$ . Figure 10 shows on the x-axis the number of phones between the DP and the end of the word; the y-axis shows the percentage recall (solid line) and precision (solid line with crosses +), respectively, while Figure 11 shows on the x-axis the number of phones between the start of the word and its DP; the y-axis again shows the percentage recall (solid line) and precision (solid line with crosses +), respectively.

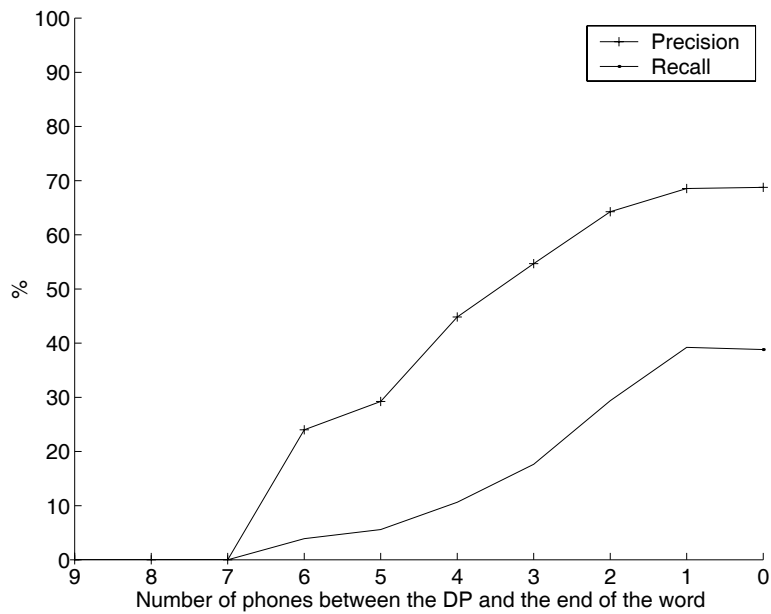


Figure 4-10. The x-axis shows the number of phones between the DP and the end of the word; the y-axis shows for a  $\theta=1.625$  and  $Act_{min}=0.5$ , the percentage recall (solid line) and precision (solid line with crosses +), respectively.

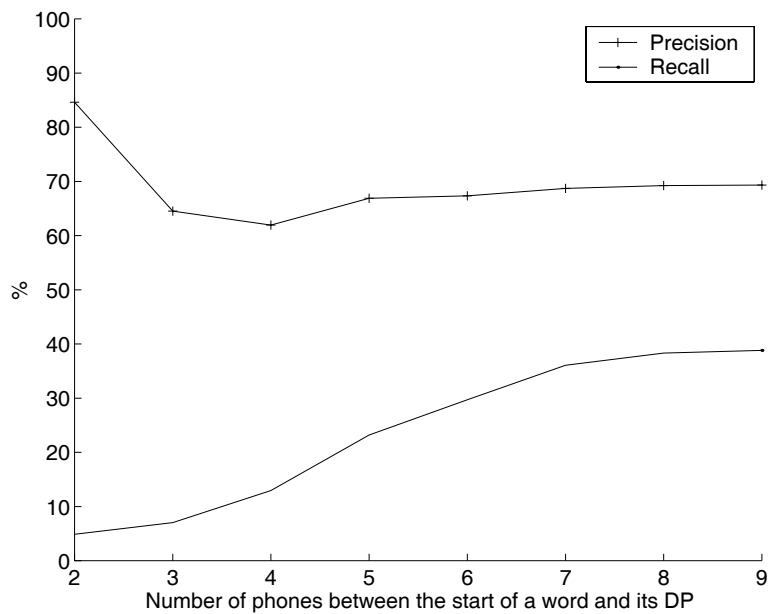


Figure 4-11. The x-axis shows the number of phones between the start of the word and its DP; the y-axis shows for a  $\theta=1.625$  and  $Act_{min}=0.5$ , the percentage recall (solid line) and precision (solid line with crosses +), respectively.

Figures 4-10 and 4-11 clearly show what was already (implicitly) shown in Figures 4-8 and 4-9, respectively. Precision and recall increase if the number of phones remaining after the DP is smaller (Figures 4-8 and 4-10). This is easy to explain, since mismatches in the part of the word that is as yet unseen cannot be accounted for in the activation measure, but the risk that future mismatches occur will be higher if more phones remain until the end of the word. At the same time, performance – in terms of recall – increases if the DP is later, so that more information in support of the hypothesis is available (Figures 4-9 and 4-11). This too makes sense, since one may expect that a high activation measure that is based on more phones is statistically more robust than a similarly high value based on a small number of phones. What should be noted, however, is that Figures 4-9 and 4-11 further suggest that precision is not dependent on the number of phones between the start of a word and its DP: The trade-off between the false accepts and the correctly recognised focus words does not change much.

#### 4.6.4 Summary

We investigated two types of predictors used for deciding whether a word is considered as recognised before the end of its acoustic realisation. The first type of predictor is related to the absolute and relative values of the word activation,  $Act_{min}$  and  $\theta$ , respectively. The results showed that the actual values of  $Act_{min}$  and  $\theta$  should not be set too high or too low, since both predictors function as filters: The higher the values for both predictors, the fewer words are recognised, and vice versa. We did not identify an optimal setting because we were not interested in optimising SpeM for a specific task.

The second type of predictor is related to the number of phones of the word that have already been processed and the number of phones that remain until the end of the word. Not surprisingly, the results showed that SpeM's performance increases if the amount of evidence in support of a word increases and the risk of future mismatches decreases. These results clearly indicate that early recognition is indeed dependent on the structure and the contents of the lexicon. If a lexicon contains many (long) words that have an early UP, decisions can be made while only little information is known, increasing the risk of errors. It is an obvious issue for follow-up research to investigate whether the decision thresholds for  $\theta$  and  $Act_{min}$  can be made dependent on the phonemic structure of the words on which decisions for early recognition must be made.

Summarising, we observed that a word activation score that is high and based on more phones with fewer phones to go predicts the correctness of a word more reliably than a similarly high value based on a small number of phones or a lower word activation score.

### 4.7 General discussion and conclusion

Human listeners are often able to reliably identify a word before the end of its acoustic realisation (Marslen-Wilson, 1987). Human listeners not only use acoustic-phonetic



information, but also contextual constraints to make a decision about the identity of a word. This makes it possible for human listeners to recognise content words even before their uniqueness point. In the research presented in this paper, we investigated an alternative ASR system that is able to recognise words *during* the speech recognition process, called SpeM, for its ability for recognising words before their acoustic offset, a capability that we dubbed ‘early recognition’. We define early recognition as the reliable identification of spoken words *before* the end of its acoustic realisation, but *after* the uniqueness point of the word (given the lexicon). The restriction to recognition at or after the uniqueness point allowed us to focus on acoustic recognition only, and minimise the impact of contextual constraints. One might wonder whether an advanced statistical language model would be able to emulate the context effects that enable humans to recognise words even before their uniqueness point. This would make SpeM’s recognition behaviour more like human speech recognition behaviour.

In our analyses, we investigated the Bayesian word activation and the contents and structure of the lexicon as predictors for early recognition. The results in Section 4.6 indicate that the Bayesian activation can be used as a predictor for the on-line early recognition of polysyllabic words if we require that the quotient of the activations of the two hypotheses with the highest scores ( $\theta$ ) and the minimum activation ( $Act_{min}$ ) both exceed a certain threshold. There is, however, a fairly high percentage of false alarms. In the subsequent analysis, we found an effect of the amount of evidence on the performance. If the DP was later in the word, thus with increasing acoustic evidence in support of a word, the performance in terms of precision and recall improved. Furthermore, the risk of future mismatches decreases with fewer phones between the end of the word and the DP, which also improves the performance. The predictors we have chosen have their parallels in the research area that investigates word confidence scores. For instance, the predictor  $\theta$  is identical to the measure proposed in Brakensiek et al. (2003) for scoring a word’s confidence in the context of an address reading system, while  $\theta$  and  $Act_{min}$  are reminiscent of the graph-based confidence measure introduced in Wessel et al. (2001). The definition of word activation in SpeM resembles the calculation of word confidence measures (e.g., Bouwman et al., 2000; Wessel et al., 2001) in that both word activation and word confidence require a mapping from the non-normalised acoustic and language-model scores in the search lattice to normalised likelihoods or posterior probabilities. Conceptually, both word activation and word confidence scores are measures related to the ‘probability’ of observing a word given a certain stretch of speech (by the human and automatic speech recogniser, respectively). However, conventional procedures for computing confidence measures only provide the scores after the end of an utterance.

The incremental search, used by SpeM to recognise a word before its acoustic offset, in combination with the concept of *word activation* proposed in this study opens the door towards alternatives for the integrated search that is used in almost all current ASR

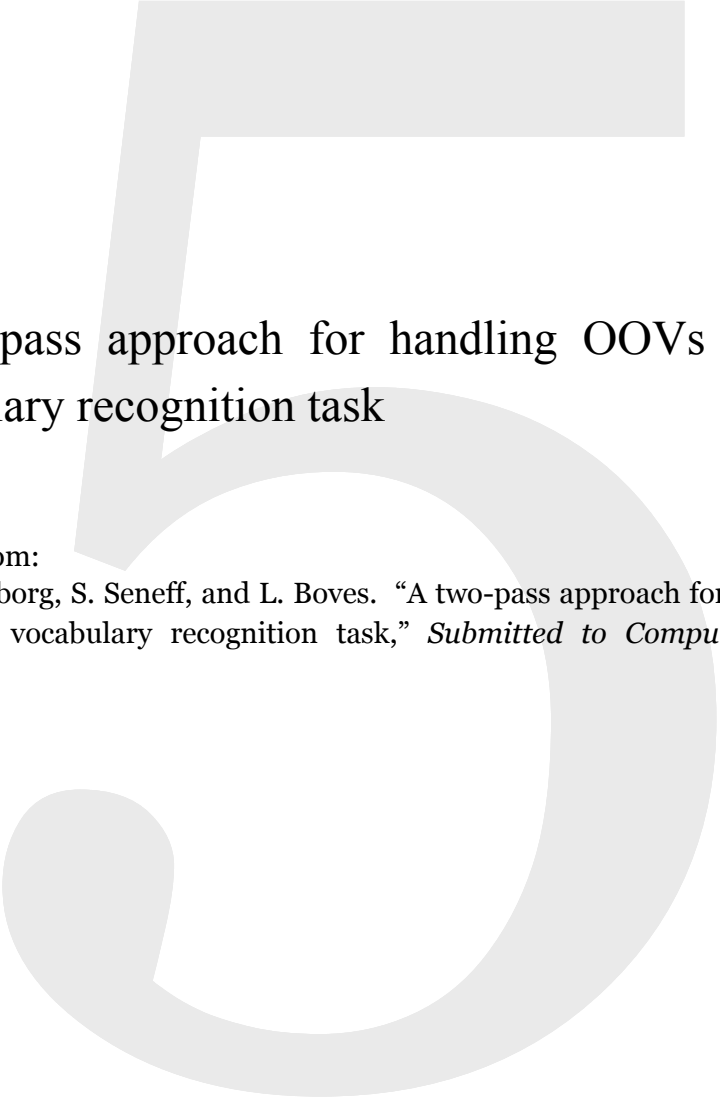
systems. An incremental search combined with word activations will be able to spot potential problems such as restarts, hesitations, and repetitions. This will be beneficial for speech-centric multi-modal interaction applications.

In conclusion, we showed that SpeM, consisting of an APR and a lexical search module, is able to recognise words before the end of the word is available. In other words, the results presented in this paper showed that early recognition in an ASR system is feasible. This property of SpeM is based on the availability of a flexible decoding during the word search and on the availability of various scores along the search paths during the expansion of the search space. The early recognition process is comparable to the early selection procedure human listeners perform while decoding everyday speech. However, there is still ample room for improvement. First, the performance of SpeM as a standard ASR system is mediocre. SpeM obtained a WER of 40.4% on a set of utterances with lengths between two and five words, while of the 1,463 focus words, 64.0% (936 utterances) were recognised correctly at the end of the utterance. We can think of several ways for improving this performance. It has been shown that optimising the performance of the APR helps to improve the performance of SpeM as an ASR system. The same holds for the addition of an N-gram language model to the lexical search module. The search can also be improved by making the insertion, deletion, and substitution penalties dependent on the identity of the phones. For example, substitutions between the phones /t/ and /d/, which differ only in one phonetic feature, could be made smaller than the substitution of /t/ for /ʃ/, where the number of different features is higher.

For 81.1% of those 936 correctly recognised focus words (51.9% of all focus words), the use of local word activation allowed us to identify the word before its last phone was available, and 64.1% of those words were already recognised one phone after the uniqueness point. However, the straightforward predictors that we derived from the Bayesian word activation appeared to be not very powerful predictors for correct decisions about the identity of a word before its acoustic completion. Yet, we are confident that the predictive power of measures derived from word activation can be improved, if only by making decision thresholds dependent on knowledge about the words that are being hypothesised. Last but not least, we believe that improvements in the APR will have a positive effect on the difference in word activation between the correct words and their competitors.

## Acknowledgements

The authors would like to thank Bart Kerkhoff, Tom Evers, Bram Vonk, and Joran Kapteijns, Computer Science students of the Radboud University Nijmegen, for their new implementation of SpeM. Furthermore, the authors would like to thank an anonymous reviewer for useful comments on an earlier version of this manuscript.



## A two-pass approach for handling OOVs in a large vocabulary recognition task

Adapted from:

O. Scharenborg, S. Seneff, and L. Boves. “A two-pass approach for handling OOVs in a large vocabulary recognition task,” *Submitted to Computer Speech and Language*.

*In this paper, we address the problem of recognising a large vocabulary of over 50,000 city names within a telephone access spoken dialogue system. The experiments are conducted on spontaneous utterances within a joint domain of two spoken dialogue systems, a weather domain (Jupiter) and a flight reservation (Mercury) domain. We adopt a two-stage framework in which only major cities are explicitly represented in the first stage lexicon. We rely on an unknown word model encoded as a phone loop to detect out-of-vocabulary (OOV) city names (also referred to as rare city names). Furthermore, we utilise SpeM, a tool that can extract words and word-initial cohorts from phone graphs on the basis of a large fallback lexicon, to provide an N-best list of promising city name hypotheses on the basis of the phone sequences generated in the first stage. This N-best list is then inserted into the second stage lexicon for a subsequent recognition pass.*

*Experiments were conducted on a set of spontaneous telephone-quality utterances from both domains. These utterances were selected because they each contained a rare city name. The first experiment showed that SpeM was able to include nearly 75% of the correct rare city names in an N-best hypothesis list of 3000 city names.*

*In addition to the N-best lists of most likely words, the lexicon of the second stage also contains the so-called 'base' lexicon (which covers the other words in the utterance). In the second recognition experiment, we tested two methods to create this base lexicon. The first method uses the same base lexicon as in the first stage, whereas the second method utilised a greatly pruned lexicon, based on the contents of the outputs of the first stage. The accuracy of the baseline recognition system (which excluded the N-best lists provided by SpeM) was 69.3%. Adding the N-best lists created by SpeM increased the accuracy to 77.3%, a relative improvement of 11.5%. While the system with the pruned general lexicon did not outperform the other system in terms of overall recognition error rate, it was able to correctly recognise up to 5% more rare city names. The final recognition results showed that about 1/3 of the rare city names that were found by SpeM were correctly recognised. So, work still remains to be done to improve on the second stage recogniser.*

**Keywords:** *automatic speech recognition; large vocabulary speech recognition; out-of-vocabulary word modelling*

## 5.1 Introduction

Jupiter (Glass et al., 1999, Zue et al., 2000) is an on-line spoken dialogue system that provides weather forecasts via a toll-free telephone number. In its current configuration, Jupiter is able to handle requests for about 500 cities. Jupiter's weather source has recently been expanded such that it can now provide weather information for 38,000 cities, which means that all 38,000 city names need to be incorporated into the speech recogniser in some way, before the additional weather information can be made available to callers. (The city names originally included in the Jupiter lexicon are referred to as 'frequent' city names, while the newly added city names are referred to as 'rare city names'.) A

straightforward solution is simply to expand the recogniser's lexicon, which will, however, result in an extremely large search space, with only a back-off (thus small) prior probability associated with each of the rare city names. Very large lexicons do not necessarily pose a problem for automatic speech recognition (ASR) systems, but the combination with a weak language model, which only has small prior probabilities associated with each word, usually results in poor performance.

To overcome the problem of a weak language model, we adopt here a novel strategy which uses small-sized lexicons in combination with a generic phone-based *unknown word* or *out-of-vocabulary (OOV) word* model to represent a rare city name (when present in an utterance) in the form of a phone sequence. This approach licenses in a second stage only those city names that match the proposed phone sequence sufficiently well (this will be explained in more detail in Section 5.2). We thus propose a two-stage recognition system with on the one hand a greatly expanded city-name capability, and on the other hand a small lexicon size, which will keep the size of the search space manageable for the real-time constraints imposed by the interactive dialogue.

The goal of this study is to build a two-stage recogniser that detects OOV words in the first stage, and adapts the lexicon of the second stage on the basis of an analysis of the phonemic composition of the OOV intervals. In the second stage, the aim is to recognise as many of the rare city names that are marked as OOV by the first stage recogniser as possible. Since an ASR system can only recognise those words that are included in its lexicon, it is clear that the performance of the second stage recogniser on recognising the OOV words is crucially dependent on whether the correct word is included in the second stage recogniser's lexicon. Optimising the coverage of the second stage lexicon is the main focus of this work (see Section 5.4).

In the literature, a variety of different solutions to handle OOV words have been proposed. These solutions can roughly be divided into two groups. In the first group (e.g., the Hypothesis Driven Lexical Adaptation (HDLA) method proposed by Geutner et al. (1999) and the Multi-pass Automatic Speech recognition uSIng Vocabulary Expansion (MASSIVE) method proposed by Ohtsuki et al. (2004)), a subset of words (to ensure that the lexicon and, thus, the search space of a second stage recogniser remain manageable) from a large fallback lexicon is selected on the basis of the results of a first stage recogniser. The selected subset is then added to the lexicon of the second stage recogniser. The second group of solutions omits a fallback lexicon, and thus other techniques have to be found to deal with the OOVs (e.g., decompounding strategies (Laureys et al., 2002); or using a phone loop as an OOV 'word' parallel to the words in the lexicon (Bazzi and Glass, 2000, 2001)).

In this research, a large fallback lexicon is available in the form of the list of city names. Therefore, in accordance with Geutner et al. (1999) and Ohtsuki et al. (2004), we built a two-stage recogniser that uses the outcome of the first recognition stage to create an

adapted lexicon for the second stage recogniser by selecting a subset from the fallback lexicon.

To select the subset of words from the large fallback lexicon, the HDLA method (Geutner et al., 1999) uses morphology, and phonetic and grapheme distances, while MASSIVE (Ohtsuki et al., 2004) measures the distance between the input speech and the words in the vocabulary database in terms of word co-occurrence patterns, in order to select the optimal subset. In this research, we use SpeM (SPEech-based Model of human speech recognition (Scharenborg et al., 2003a, 2003b)) – a tool originally designed for the simulation of human speech recognition (HSR) processes. SpeM is used to extract words and *word-initial cohorts* (words sharing phone prefixes) from the fallback lexicon on the basis of the phonemic distances between the phones in a phone graph and the phonemic representation of the words and word-initial cohorts in the fallback lexicon (see Section 5.2.2).

This research is part of a larger research project aiming at providing seamless domain switching among multiple domains within a single conversational agent. Towards that goal, we have combined the vocabularies of two pre-existing systems, the Jupiter system in the weather domain (Glass et al., 1999; Zue et al., 2000) and the Mercury system in the flight domain (Seneff, 2002). There is a large overlap in the general lexicons of the Jupiter and Mercury systems. For instance, they share general question syntax and dates, as well as city names, making it a logical step to combine the two systems into one domain-independent system. We have included the original set of 500 major cities in the lexicon of the first stage recogniser. We hope that a rare city name uttered by the speaker will appear as an unknown city in the  $N$ -best list of the first stage.

## 5.2 The proposed two-stage recognition system

The proposed two-stage recognition system is schematically depicted in Figure 5-1. The acoustic signal is fed into the first stage recogniser, which uses a lexicon that captures ‘general’ words (see Section 5.3.2 for more details) in addition to the 500 most frequent city names. Since the method we propose to deal with OOV words in a two-stage recognition system is crucially dependent on the detection of the OOV intervals by the first stage recogniser, an OOV model that is intended to mark all city names not in the lexicon as being OOV is integrated into the first stage (see Section 5.2.1). The hypothesised phone graphs underlying the stretches of speech signal marked as OOVs can be extracted. These phone graphs (referred to as OOV phone graphs hereafter) are used by the SpeM module to select the most likely city names from the fallback lexicon for that specific utterance. This subset of most likely city names is then added to the ‘utterance-dependent’ lexicon of the second stage (see Section 5.4). The second stage recogniser then makes a new recognition attempt on the basis of the same acoustic models as were used in the first stage. In the second stage, the lexicon (and thus the recogniser) is tuned to the utterance (and thus the domain).

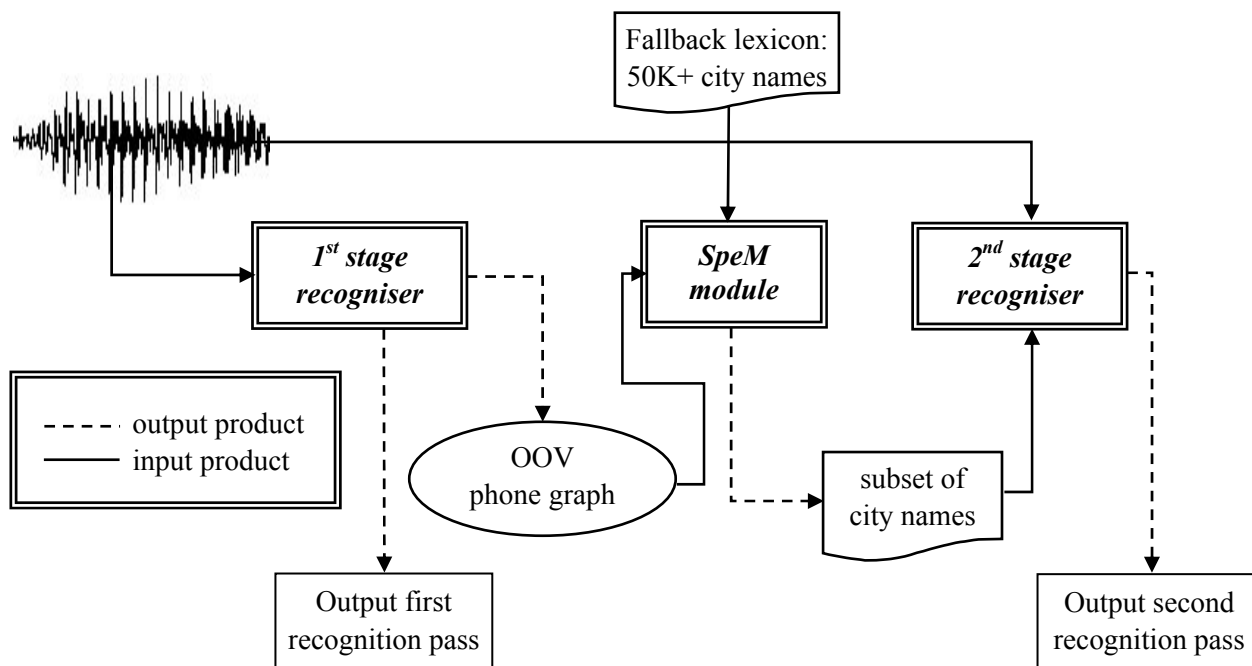


Figure 5-1. Overview of the proposed multi-stage recognition system.

### 5.2.1 Automatic speech recognition system

The two-stage recogniser used in this study is the segment-based automatic speech recognition system SUMMIT (Glass, 2003), which uses Finite State Transducers (FSTs) to represent its search space. The entire linguistic search space in the recogniser can be represented by a single FST ( $U$ ). Generally,  $U$  is represented by a cascade of FST compositions:

$$U = C \circ P \circ L \circ G \quad (5-1)$$

where  $C$  contains the mapping from the diphone labels used in the acoustic model set to monophone labels;  $P$  applies phonological rules;  $L$  maps the words in the lexicon to their phonemic representations; and  $G$  is the language model (LM).

The two-stage recogniser uses class bigram (in the forward pass) and trigram (in the backward pass) LMs (see Seneff et al. (2003) for details of the procedure.). The bigram and trigram LMs were trained on 167,967 utterances (on average 5.8 words per utterance) from both the Jupiter and the Mercury domain, and were used both in the first and the second stage recognisers (thus in contrast to the lexicons, the LMs of the second stage recogniser are not tuned to the utterance as is the case for the lexicons). In the material used for training the LMs, all rare city names were marked, and were treated as one class. Likewise, all frequent city names were treated as one class as well. The number of rare city names in

the training material was rather low; this would result in a very small probability for the rare city class. To increase the number of rare city names, 3,000 frequent city names in existing utterances were artificially changed into an unknown city name. In this way, frequent and rare city names were treated as different classes and separate LM scores were calculated for them. The frequent city names have their own unigram scores within the frequent city class; the rare city names in the rare city class, on the other hand, all have equal probability (see also Section 5.2.1).

### *Detecting the OOVs*

The procedure used to mark the OOV words and generate the OOV phone graphs is described in detail in Bazzi and Glass (2000, 2001): The *generic word model* is implemented as a phone  $N$ -gram (a phone loop that allows for phone sequences of arbitrary length). This OOV model is included in the lexicon ( $L$ ) and as such is wired into the linguistic search space ( $U$ ). The transition into the generic word model is controlled via an OOV penalty. This OOV penalty can be considered as a unigram score: It controls how easily the OOV ‘word’ is selected.

Underlying the hypothesised OOV is the OOV phone graph. For each utterance, in which an OOV was hypothesised in the word lattice, only one OOV phone graph was generated (this is due to the implementation of the procedure to extract the OOV phone graphs). This means that where an utterance contains more than one OOV, as in *I'd like to fly from <OOV> to <OOV>*, an underlying OOV phone graph can be generated for only one of the OOVs. In this experiment, this is not a problem since, in the test data used in this study, each utterance contained at most one rare city name.

Note that an OOV might be hypothesised for a stretch of the speech signal that does not correspond to a city name. Furthermore, it is possible that the phone graph does not match exactly with the stretch of speech that contains the rare city, i.e., it is possible that additional phones are included at the start or the end of the phone graph that do not belong to the rare city name. Likewise, it is possible that the word is truncated, i.e., phones of the rare city name are missing at the start or the end of the phone graph. Finally, it is possible that the first stage recogniser incorrectly recognises the rare city name as an in-vocabulary word. In this study, whenever the first stage recogniser incorrectly recognised the rare city name as an in-vocabulary word, those data (less than 5% of the total) were removed from the test set.

### *The ‘dynamic’ lexicon*

The recognisers in the first and second stage are identical, with the exception of the lexicon ( $L$ ) and the prior probability of the OOV model. The lexicon of the second stage consists of three components, as shown in Figure 5-2. The first two components, the ‘Base’ lexicon and the OOV ‘word’ model are also present in the lexicon of the first stage recogniser. The



new ‘dynamic’ lexicon can be wired on-the-fly into the search space of the recogniser, without the need to rebuild the lexical search space (Chung et al., 2004). This dynamic model is supplied with the list of rare city names extracted by SpeM from the fallback lexicon (see also Sections 5.2.2 and 5.3). Our city name lexicon contains 52,595 city names, which were harvested from the World Wide Web. Most of the cities were non-existent in our lexical baseforms resource file, and pronunciations were therefore automatically generated for them using the letter-to-sound system described in Chung et al. (2004) and Seneff (2004). The errors in these pronunciations have not been corrected manually. This further challenges the recognition task. The words in the dynamic lexicon have identical unigram scores. This is in contrast to the words in the base lexicon which all have their own unigram scores. The probability of choosing a word from the dynamic lexicon is controlled via the previously described language model scores for the rare city name *class*.

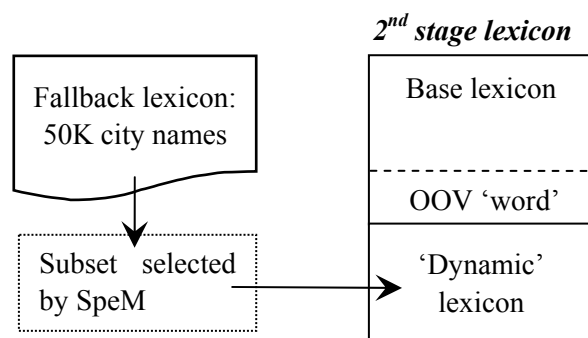


Figure 5-2. The components of the utterance-dependent lexicon of the second stage recogniser.

The OOV model is included in the lexicon of the second stage recogniser, because also in the second stage OOV words can occur, and we want to be able to correctly deal with them. The language model scores of the OOV model in the second stage, however, are smaller than the language model scores of the OOV model in the first stage, since in the first stage recogniser an OOV should be hypothesised more often than in the second stage.

### 5.2.2 SpeM

SpeM was originally implemented to serve as a tool for research in the field of human speech recognition. It is a new and extended implementation of the theory underlying the *Shortlist* model, a computational model of human word recognition developed by Norris (1994). The main advancement of SpeM over pre-existing computational models of human word recognition is that SpeM uses the acoustic speech signal as input, while *Shortlist* and other computational models of HSR only take handcrafted symbolic representations (e.g., phonemes or linguistic features) as input. Furthermore, SpeM supports unigram and bigram language models, while the *Shortlist* and the other HSR models do not support language

models. Besides its use as a tool for simulating results found in HSR experiments, SpeM can also function as an experimental ASR system, e.g., for the recognition of words before the end of the acoustic realisation of the word is complete (Scharenborg et al., 2003c, 2004).

SpeM consists of two modules: An *automatic phone recogniser* (APR) and a *word search module*. The word search module parses the probabilistic phone graph created by the APR in order to find the most likely (sequence of) words, and computes for each word its activation based on the accumulated acoustic evidence for that word (Scharenborg et al., 2003a, 2003b). In the experiments described in this paper, the phone graphs are created by the first stage recogniser. In the remainder of this paper, whenever the word ‘SpeM’ is used, this actually only refers to the word search module of SpeM – unless stated otherwise.

In SpeM, the sequence of words with the smallest phonemic distance between the sequence of phones on the path through the OOV phone graph and the phonemic representations of the words in the fallback lexicon (represented as a lexical tree) is determined using a time-synchronous and breadth-first dynamic programming (DP) algorithm. Each phone insertion, deletion, and substitution is penalised according to independent penalties which can be tuned separately (for more details the reader is referred to Scharenborg et al. (2003b)). Furthermore, a garbage phone model is included in the lexicon. This garbage phone model is mapped onto phones appearing at the start and at the end of the phone graph that belong to the preceding and following word. The output of SpeM consists of an  $N$ -best list of hypothesised parses. Each parse contains words, word-initial cohorts, garbage, silence, and any combination of these, with the exception that a word-initial cohort can only occur as the last element in the parse. Thus, in addition to recognising full words, SpeM is able to recognise partial words. The parameter settings of SpeM are such that, although SpeM is able to recognise sequences of words, the recognition of a single word (preceded or followed by garbage) is more likely.

Subsequently, if a word-initial cohort has been recognised for an utterance (or OOV phone graph), and if it consists of more than three phones, the word-initial cohort is ‘unpacked’: All words belonging to the word-initial cohort are listed. Finally, for each utterance (or OOV phone graph), the top  $N$  words in the output of SpeM form an utterance-specific  $N$ -best list that goes into the utterance-specific dynamic lexicon of the second stage recogniser. The effect of the size of these  $N$ -best lists is investigated in Section 5.4.

Note that SpeM always returns an  $N$ -best list of most likely city names, even if the phone graph did not correspond to a city name. This is because SpeM searches a lexicon that contains only city names.

## 5.3 Experimental set-up and materials

### 5.3.1 Experimental set-up

In the first series of experiments (Section 5.4), we focussed on the selection of the rare city names out of the fallback lexicon. If the second stage recogniser's lexicon does not contain a particular word, it is impossible for the second stage recogniser to recognise it. The aim for SpeM is thus to find the correct rare city name for as many utterances as possible in the fallback lexicon. We tested two variables: 1) the size of the utterance-dependent  $N$ -best lists generated by SpeM; 2) the effect of an utterance-dependent language model that boosts the probability of the city names in a given (US) state, if a state name was recognised by the first stage recogniser.

The results of this experiment are presented in terms of coverage, i.e., the percentage of the test set utterances for which the rare city name in its transcription (which was presumably marked as OOV by the first stage recogniser) is present in the  $N$ -best list generated for that utterance by SpeM.

In the second series of experiments (Section 5.5), the  $N$ -best lists generated by SpeM were included in the dynamic lexicon of the second stage recogniser. Besides the dynamic lexicon and the OOV model, as is shown in Figure 5-2, the lexicon of the second stage also includes the base lexicon. The base lexicon can be created in different ways. We compared the approaches of Ohtsuki et al. (2004) and of Tang et al. (2003). Both include a pruning stage based on the first stage hypotheses. The results of this series of experiments are presented in terms of word accuracy.

### 5.3.2 Materials

The experiments were conducted on a set of continuous speech utterances, recorded from interactive telephone conversations with both the Mercury and the Jupiter systems. The independent test set consisted of 418 utterances taken from both domains, each utterance containing exactly one rare city name. The first stage recogniser did not detect an OOV in 19 utterances of the test set (4.5%), which means that the rare city name was recognised incorrectly as an in-vocabulary word. If no OOV is detected, no OOV phone graph is generated, and SpeM will not be able to extract a list of words from the fallback lexicon for those utterances. Consequently, no improvement of the recognition of the rare city names by the second stage recogniser can be expected. These utterances were discarded from the test set, leaving 399 utterances that were used in the experiments.

The lexicon of the first stage consisted of the 'general' words from both domains, a short list of the 500 most frequent city names, all US state names, and a set of 1,326 partial and short city names with a phonemic representation of three phones or less, such as 'los', 'ann', 'new' – this to simplify SpeM's task, since short words are difficult to find in a phone lattice. This resulted in a lexicon of 2,802 words. Note that other OOV words

besides city names can occur. In our complete test set, ten words (in nine utterances) other than rare city names were missing from the first stage recogniser's lexicon. In only one of those nine utterances, the phone graph that was extracted did not correspond to the phone graph underlying the rare city name. In that case, SpeM is of course unable to find the correct rare city name.

## 5.4 Extracting the subset from the fallback lexicon

We first investigated the size of the utterance-dependent  $N$ -best lists. Does the coverage increase proportionally with the increase of the size of the  $N$ -best list? Or, does SpeM behave similarly to standard beam search techniques, in which a small beam width is already able to maintain the hypotheses that are most promising and to suppress the hypotheses that are unlikely, such that broadening the beam width does not improve performance much (Ney & Ortmanns, 2000)?

### 5.4.1 No utterance-dependent language models

The size of the utterance-dependent  $N$ -best lists created by SpeM was varied between 500 and 3000 in steps of 500 entries. The results are shown in Table 5-1. The second column (denoted 'No LM') shows the coverage for the varying sizes of the  $N$ -best lists in terms of absolute number of utterances for which the correct rare city name was present in the  $N$ -best list ('Abs') and as a percentage of the total number of 399 utterances of the test set ('%'). For these results, SpeM was run with no utterance-dependent language models.

*Table 5-1.* Coverage results and analysis for varying sizes of the  $N$ -best lists generated by SpeM.

N-best list size	No LM		Max. Gain Possible	With LM		Analysis	
	Abs	%		Abs	%	Loss	Gain
500	223	55.9	102	275	64.4	6	59
1000	230	57.6	98	281	70.4	7	58
1500	234	58.6	95	284	71.2	9	59
2000	235	58.9	95	291	72.9	9	65
2500	238	59.6	93	293	73.4	11	66
3000	239	59.9	92	296	74.2	10	67

The coverage results show that over 55% of the rare city names that were missing from the lexicon of the first stage recogniser, and thus could not be recognised in the first stage, are now present in the lexicon of the second stage. This is an encouraging result, bearing in mind that all 52,595 words in the fallback lexicon have equal probability, and that the

generated OOV graphs are far from perfect (because of the possibility of preceding and trailing garbage phones, as well as the possibility of phone recognition errors within the city name itself, and the possibility that the phone graph is cut off prematurely). Comparing the coverage for the  $N$ -best sizes 500 and 3000 clearly shows that increasing the length of the  $N$ -best list 6-fold does not increase the coverage with the same amount: only 16 more correct rare city names were found when the  $N$ -best list size was 3000. This shows that the selection method without LM of SpeM (the lexical search) works in a comparable way to the beam search used in standard ASR systems.

#### 5.4.2 Adding utterance-dependent language models

It might be possible to improve on the results shown in the ‘No LM’ column in Table 5-1 if city names that are more likely on the basis of the context of the utterance receive a higher probability, while the more unlikely city names are penalised. An obvious cue is the state name. It is highly likely that a city name, which is uttered in the same utterance as a state name, lies in that state. Thus, if a state name has been recognised by the first stage recogniser, that information can be used to increase the prior probability of all city names in that state. The database with state-city name information that we have available shows that on average there are 1,890 city names per state. So, if the state name were known, it would reduce the number of possible city names considerably.

In this second experiment, we built utterance-dependent language models for SpeM for those utterances in which a state name was present. If a state name is present in the  $N$ -best list generated by the first stage recogniser, all city names in that state receive a higher unigram score by adding a constant to the unigram score. In our test set, for 74.9% (299) of the 399 utterances, a state name is present. For 243 utterances, a state name appeared in the  $N$ -best lists generated by the first stage recogniser. For these utterances, utterance-dependent language models were created. Of course, the 56 utterances in which a state name was present but not found by the first stage recogniser and the 100 utterances in which no state name was present will not benefit from this approach.

First, we tabulated for how many of the 243 utterances for which a state name was recognised by the first stage recogniser, the correct rare city name was present in the  $N$ -best list generated by SpeM. In this way, the maximum gain could be calculated. The column ‘Max. Gain Possible’ in Table 5-1 contains the number of utterances in which the first pass recogniser found a state name, and in which SpeM could not find the correct city name in the first experiment.

The fourth column denoted ‘With LM’ shows the coverage in terms of absolute number of utterances (‘Abs’) and the percentage of the full test set (‘%’) when the utterance-dependent language models were added to SpeM. As can be seen from this column, there is a clear increase in coverage. In the case of an  $N$ -best list of 500, the rare city name was

selected into the utterance-dependent lexicon for 52 more utterances; for an  $N$ -best list of 3000, this number is slightly larger: 57, resulting in a coverage of 74.1%.

Furthermore, comparing the ‘No LM’ with the ‘With LM’ column shows that the difference in coverage between the 500- and 3000-best lists in the ‘No LM’ case is 4.0%, while when an LM is available this difference is larger, viz. 9.8%. This difference in increase clearly shows that the utterance-dependent language models created using state information are doing well in directing the search in SpeM: even rare city names that were very difficult to find at first without an LM, are now found more often. Adding additional words to the  $N$ -best lists, as is done when using a 3000-best list instead of a 500-best list, now makes it possible to find these difficult-to-find rare city names.

### 5.4.3 Analysis and discussion

There is always the risk that the state name recognised by the first stage recogniser is incorrect, or that another (major) word in the utterance is incorrectly recognised as a state name, which will result in a boost of the probability of the wrong city names. Therefore, we also analysed the number of utterances for which adding the language models made the correct rare city disappear from the utterance-dependent  $N$ -best list by SpeM. These results are shown in the column ‘Analysis’ in Table 5-1. For instance, in the case of an  $N$ -best list of length 500, for 6 utterances for which the correct rare city had been included in the  $N$ -best list in the case that no LM was used, the correct rare city name was no longer selected when the LMs were used. On the other hand, for 59 utterances for which the correct city name was missing from the  $N$ -best lists generated by SpeM, the correct rare city names were selected once the LMs were used.

In conclusion, adding the utterance-dependent language models to SpeM resulted in a net gain in coverage compared to the situation when no language model was used: for 2/3 to 3/4 (depending on the size of the  $N$ -best list created by SpeM) of the test set, the rare city name that was missing from the first stage lexicon and therefore could not be recognised by the first stage recogniser is now included in the lexicon of the second stage recogniser. This will increase the probability that the second stage recogniser ultimately recognises the correct rare city name. In the recognition experiments discussed in the next section, we always used the  $N$ -best list generated by SpeM with the LM.

## 5.5 Performance of the two-stage recogniser

For each of the utterances in the test set, an utterance-dependent  $N$ -best list that can be added to the dynamic lexicon of the second stage recogniser is now available. The full system, with the base lexicon and the  $N$ -best lists generated by SpeM, is used to run a full recognition attempt for each utterance of the test set. The performance of the recognition system is measured in terms of accuracy (of all words in the test set, not just the rare city names), but since we are mainly interested in the recognition of the rare city names, the number of correctly recognised rare city names will also be presented.

What remains is the construction of the base lexicon (see Figure 5-2). We investigated two methods to construct the base lexicon: one using the same base vocabulary as in the first stage, and the other using a much smaller lexicon derived directly from the output of the first stage recogniser.

To understand how well the recognisers are able to perform once the *N*-best lists are wired into the dynamic lexicon, ‘measured upper-bound’ (M.U.B.) performance was calculated for both types of recognition system. To that end, the rare city names in the reference transcriptions were substituted by OOV. The test set was then recognised with a system that uses the baseline lexicon plus the OOV-word. By doing so, we created a task that effectively had no OOV. The ‘measured upper bound’ is the accuracy obtained with these systems. We then compared the performance of the systems with OOVs with this M.U.B. baseline.

### 5.5.1 The Ohtsuki method

The easiest way to construct the base lexicon is to use the same lexicon as was incorporated into the first stage recogniser. The dynamic lexicon is then in effect an addition to the first stage recogniser’s lexicon. This method has also been used by Ohtsuki et al. (2004) for a similar task. The column denoted ‘Ohtsuki method’ in Table 5-2 presents the results for varying sizes of the *N*-best lists generated by SpeM. The size of ‘0’ is the baseline result: in this case, only the base lexicon (including the OOV model) is used for recognition. The baseline system is thus identical to the first stage recogniser.

Table 5-2. Results of the two-stage recogniser for varying sizes of the *N*-best list generated by SpeM, and two different methods to construct the base lexicon.

<b><i>N</i>-best list size</b>	<b>Ohtsuki method</b>			<b>Tang method</b>		
	<i>Acc. (%)</i>	<i>#rare cities</i>	<i>Lex. size</i>	<i>Acc. (%)</i>	<i>#rare cities</i>	<i>Lex. size</i>
<i>M.U.B</i>	73.4	0	2,802+1	77.1	0	± 23.5+100+1
0	68.3	0	2,802+1	69.3	0	± 23.5+100+1
500	<b>77.3</b>	101	2,802+501	76.9	<b>106</b>	± 23.5+100+501
1000	77.0	97	2,802+1001	<b>77.3</b>	104	± 23.5+100+1001
1500	76.8	96	2,802+1501	77.1	101	± 23.5+100+1501
2000	76.7	95	2,802+2001	77.1	101	± 23.5+100+2001
2500	76.7	93	2,802+2501	76.9	100	± 23.5+100+2501
3000	76.4	93	2,802+3001	76.8	100	± 23.5+100+3001

As shown in Table 5-2, the system with an  $N$ -best list size of 0 has an accuracy of 68.3%; adding an  $N$ -best list with the 500 most likely city names already increases the accuracy by 9.0 percentage points, while 101 rare city names are recognised correctly. The measured upper-bound ('M.U.B.' in Table 5-2) accuracy is 73.4% when Ohtsuki's method is used to construct the base lexicon. What is striking is that adding an  $N$ -best list with the 500 most likely city names increases the accuracy beyond the measured upper-bound accuracy. We will discuss this in more detail in Section 5.5.3. Further increasing the lexicon size, however, improves neither the accuracy nor the number of correctly recognised rare city names. The latter is most likely due to the similarity of the words in the  $N$ -best lists generated by SpeM. The words are similar because they are the most likely words from the fallback lexicon given the phone graph corresponding to the OOV stretch. This increases the confusability of the words in the lexicon, which in turn puts a curb on the maximum accuracy that can be obtained. We therefore sought a way to decrease the size of the base lexicon.

### 5.5.2 The Tang method

Tang et al. (2003) demonstrated that, for a two-stage recognition system, the most important words to retain in the lexicon of the second stage are the in-vocabulary words that have been recognised by the first stage recogniser, augmented with a list of those words that are most often deleted by the first stage recogniser. These words are usually short function words such as 'a', 'the', 'to', etc. Although the two-stage recognition system designed by Tang et al. has a different goal (it is used for the recognition of the sub-word linguistic features 'manner' and 'place of articulation') than the two-stage recognition system presented in this study, we think that Tang's method might be useful to decrease the size of the second stage lexicon in comparison with the method presented in the previous section.

The base lexicon in this second experiment consisted of:

- The 100 words that were most often deleted by the recogniser in the first stage.
- All words in the 50-best list created by the recogniser in the first stage. This resulted in on average 23.5 different words for each utterance.

The results are presented in the 'Tang method' column in Table 5-2. The lexicon size is reduced dramatically when Tang's method is used (compare the two columns denoted 'Lex. size'). The baseline system ( $N$ -best list size == 0) shows a higher accuracy than the baseline system when the base lexicon was made following Ohtsuki's method. This indicates that the internal confusability within the base lexicon has decreased, due to the decrease in the size of the base lexicon.



For the Tang method, the measured upper-bound accuracy is 77.1% (see ‘M.U.B.’ row, Table 5-2). In the case Tang’s method is used, the best accuracy is obtained when 1000 of the most likely words are added to the dynamic list. As can be seen in Table 5-2, this accuracy is equal to the best accuracy obtained with the Ohtsuki method. However, the number of correctly recognised rare city name differs. When using Tang’s method, the number of correctly recognised rare city names is higher than with Ohtsuki’s method. Again, the best performance exceeds the measured upper-bound accuracy (see also Section 5.5.3).

The highest number of correctly recognised rare city names, however, is obtained for an  $N$ -best list size of 500 with the Tang method. In this case, 106 of the rare city names have been recognised correctly, as contrasted with only 101 for the equivalent  $N$ -best size using the Ohtsuki method. Comparing the results of the 500-best list and the 1000-best list recognition systems revealed that the lower accuracy of the former is due to a higher number of insertions.

### 5.5.3 Analysis and discussion

The measured upper-bound accuracies for the Ohtsuki method and the Tang method differ (Ohtsuki: 73.4% vs. Tang: 77.1%). This difference is due to the fact that, in the latter case, an OOV was hypothesised 90 more times (235 times in total), which will of course more often result in a ‘hit’. More interestingly, however, both methods have a best performance that exceeds the accuracies obtained by collapsing all rare city names to the OOV word (the measured upper-bound). It is known from the literature (e.g., Gauvain et al., 1994; Hetherington, 1994, 1995) that if a stretch of speech cannot be mapped onto a lexical item, the segmentation of that stretch of speech into words is hampered. Once the rare city name is added to the lexicon, and thus the OOV problem is removed, the segmentation of that stretch of speech is much easier and fewer errors are made. Thus, adding the rare city names to the second stage lexicon improves not only the recognition of those rare city names but also of the words in the close proximity of those rare city names.

## 5.6 Conclusion and future work

In this work, we presented a two-stage recognition system for handling OOVs in a large vocabulary speech recognition task. We showed that SpeM is able to retrieve nearly 75% of the rare city names from a large fallback lexicon resulting in an increase of the performance of the two-stage recognition system, once the rare city names selected by SpeM were added to the lexicon of the second stage, of 8.0% and 9.0% respectively for the two types of base lexicons we used. These results are remarkable, keeping in mind that all words in the dynamic lexicon have equal probability.

The fact that SpeM was able to find the correct city name, given a phone graph, is encouraging. However, it is evident that there is substantial room for improvement. One

way to do this would be using population statistics to compute unigram probabilities for the city names, instead of a uniform value.

The final recognition results showed that about 1/3 of the rare city names that were found by SpeM were correctly recognised; thus work still remains to be done to improve on the second stage recogniser. First of all, the unigram scores of the words in the dynamic list could be improved upon. Here too, population statistics could be mapped to a unigram probability model. Secondly, the language model of the second stage is kept identical to the language model used in the first stage, while the lexicon of the second stage is tailored to the utterance. The performance of the second stage recogniser might improve more when an utterance-dependent language model is adopted. A first experiment would involve running parallel language models for the two domains – Mercury and Jupiter – in the second stage, and weighting them according to the likelihood of each domain, given consideration of word sequences extracted from the first stage.

In the deployed system, a possible way to deal with unrecognised rare city names would be to incorporate a speak-and-spell module, such as the one described in Chung et al. (2003). Whenever an OOV is hypothesised by the recognition system, the user would be guided through a speak-and-spell subdialogue to resolve the city name.

### Acknowledgements

Part of this work was carried out while the first author was visiting the Spoken Language Systems Group, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. The first author would like to express her appreciation for the hospitality of MIT and the scholarship (from the Katrien van Munster fund) awarded by the Radboud University Nijmegen; without both the visit would not have been possible. Furthermore, the authors would like to thank Chao Wang and Lee Hetherington for their help building the recognition system, and Min Tang for providing the set of test utterances.

A large, light gray number '6' is centered on the page, serving as a background for the text.

General discussion and concluding remarks

## 6.1 General discussion

In this thesis, I have attempted to narrow the gap that has existed for decades between the research fields of human speech recognition (HSR), more specifically the field of human *word* recognition, and automatic speech recognition (ASR). In Chapters 2 and 3, I focussed on the contribution of ASR for HSR. In Chapters 4 and 5, I looked more at what a computational model of HSR, called SpeM, could accomplish and contribute to ASR.

### 6.1.1 Towards an end-to-end model of human speech recognition

In Chapter 2, a first attempt to bridge the gap between the two research fields was made by creating an end-to-end model of human word recognition that was based on two existing models of both sides, namely an automatic phone recogniser (APR) and Shortlist (Norris, 1994). The experiments described in Chapter 2 illustrated the consequences of some of the simplifying assumptions made in Shortlist and other HSR models, and showed the extent to which these assumptions need to be revised to produce end-to-end HSR models that are able to deal with real-speech input. The biggest shortcoming of the joint model of the APR and Shortlist proposed in Chapter 2 was that it made ‘hard’ decisions at the level of input phones. Shortlist requires a single string of phone symbols as input. This implies that the APR is forced to make ‘hard’ decisions about the segmental representation of the speech signal based only on the acoustic information. The second shortcoming of Shortlist is that the search in the Shortlist module of the joint model is a simple lexical look-up, which causes a misalignment of all subsequent phones in the case of a phone insertion or deletion. The experiments clearly showed that a straightforward combination of the APR and Shortlist did not yield an end-to-end model of HSR that could deal satisfactorily with real-life input, even though Shortlist is a successful model for a specific aspect of the human speech recognition process.

In our search for a computational end-to-end model of human speech recognition, the research described in Chapter 2 was taken one step further in Chapter 3. The computational-level analysis (Marr, 1982) of the word recognition process made the close parallels between HSR and ASR explicit. The computational parallels were further illustrated by the development of SpeM: a computational model of HSR, based on Shortlist, that was built using techniques from ASR. SpeM is not just a re-implementation of Shortlist; it represents an important advancement over existing models of HSR in that it is able to recognise words from acoustic speech input at reasonably high levels of accuracy, while currently existing models of HSR almost invariably assume a (error-free) symbolic representation of the acoustic signal as input. In SpeM, the ‘hard decisions problem’ at the input level was solved by representing the speech signal as a probabilistic phone lattice

containing multiple phone-string hypotheses. This allows, in a natural way, the postponement of a hard decision to a point later in the word search process. The second shortcoming of the combination of the APR and Shortlist, the implementation of the lexical search, is solved in SpeM by using a search algorithm based on dynamic programming techniques that tolerates misalignments between the input and canonical phonemic lexical representations (at a certain cost). The experiments described in Chapter 3 showed that SpeM strongly outperformed Shortlist in its ability to recognise words from real-life speech, spoken by a large number of different talkers in a noisy environment, largely due to the phone-lattice representation of the input in SpeM.

According to HSR theory, words that overlap in time compete with each other through lexical inhibition during the human speech recognition process. Shortlist is a localist connectionist model, with separate nodes for each candidate word involved in the current lexical search. SpeM, on the other hand, uses *path*-based scores. In Chapter 3, we were able to show that it is possible to model lexical competition using SpeM's path-based scores. This issue is further addressed in Section 6.1.4.

#### *Advantages of an end-to-end model of HSR*

HSR modelling has tended to avoid detailed analysis of the problems of speech recognition given real speech input. The fact that HSR models cannot recognise real speech makes it hard to evaluate the theoretical assumptions underlying those models. For example, Shortlist makes the simplifying assumption that the word recognition process receives a sequence of discrete phonemes as input. This raises the question whether the theory underlying the model (prelexical and lexical level of speech processing, only a feed-forward flow of information, competition of time-overlapping words, a word can start at any point in time; see also Chapters 1 and 3) would remain valid if that simplifying assumption were abandoned. Thanks to the implementation of SpeM this could be tested. The experiments presented in Chapter 3 showed that the theory underlying Shortlist still holds when the simplifying assumption of the symbolic phone string has been removed.

Another clear advantage of end-to-end computational models of HSR is that they can be tested with precisely the same stimulus materials as used in the behavioural studies being simulated, while for older HSR models some idealised form of input representation had to be used. However, since SpeM's performance in terms of percentage correctly recognised words is far worse than human performance, it may be necessary in HSR modelling to continue to use idealised inputs, in parallel with real-speech simulations in models such as SpeM. We should note, however, that for human beings to understand each other, 100%

word recognition is not needed; at the same time, 100% word recognition is not achievable for ASR systems.

### *The performance of the APR*

One might argue that the performance of SpeM as a speech recogniser might have been better if an input had been used that is much ‘better’ than the one provided by the current APR. In all cases where SpeM would have been confronted with a realistic one-dimensional representation of the speech signal as input, the answer would undoubtedly be ‘no’. First of all, as argued in Chapter 1, it is impossible to generate a unique ‘true’ representation of the speech signal. And secondly, as argued in Chapter 2 on the basis of work by Cucchiaroni et al. (2001), if the input of a human ‘phone recogniser’ had been used, the results would not have been radically different, despite the fact that human transcriptions would have looked much better than the output of the APR. It is thus not sufficient to try to improve the single-best output of the APR.

But SpeM uses a probabilistic phone graph as input instead of a single-best one-dimensional representation of the speech signal. Will its performance be any different, even better, when a better APR will be used as front end? The answer to this question is probably ‘yes’ if an APR can be created that is able to get the number of ‘correct’ phones as high as possible in the phone graph. However, this is not the full solution to the problem. Without top-down information (Shortlist and SpeM are autonomous models, they only have a feed-forward (or bottom-up) flow of information), the phonemic transcriptions of the words are unknown. This makes it impossible to know for the APR whether it selected the correct phones into the phone graph. In order to improve the performance of SpeM, a search mechanism is needed that is able to deal with paths through the phone graph in which phones are inserted or deleted in comparison with the canonical phonemic transcriptions of words. One challenge for the future will therefore be to establish whether the limitations of SpeM’s APR can be overcome, such that phone graphs are being generated in which as many of the ‘correct’ phones are as high as possible in the phone graph.

### **6.1.2 Incremental vs. integrated search**

Most mainstream ASR systems use some kind of integrated search algorithm, while humans compute an on-line activation measure for words as the speech comes in (and presumably make a decision as soon as the activation of a word is high enough). In order to model the human speech recognition process, computational models of HSR should thus be able to provide word activation scores over time, as the input comes in. So, it should be possible to recognise words before their acoustic realisation is complete. SpeM does

exactly that: it gives a ranked list of the most likely words at each point in time while the input comes in, and thus is able to identify words before their acoustic offsets.

### *Out-of-vocabulary words and phone strings*

The incremental recognition capability allows SpeM to handle issues that conventional ASR systems have difficulty dealing with. Conventional ASR systems are only able to recognise the words that are included in their lexicon. When encountering an out-of-vocabulary word (OOV; thus a word that is not present in the lexicon), the ASR system will match the word to one of the items in its lexicon. This, of course, causes recognition errors. For an ASR system to be able to detect the phone sequences associated with OOVs, it needs to be tuned to the task of OOV detection. However, there is always the risk that an in-vocabulary word is incorrectly marked as an OOV. A bigger problem, however, is when an OOV is not recognised as such, and is thus mapped onto an in-vocabulary item.

A second type of problem for ASR systems is related to phone strings associated with garbage, non-words, and speech-like sounds. ASR systems often have included a garbage model with which they can model these phenomena. In this way, they are able to skip over phone sequences caused by truncated words, hesitations, etc. In psycholinguistics, a non-word is a phone sequence that complies with the phonotactic constraints of a language, but is not an existing word in the language, e.g., *ploem* for Dutch or *fourf* for English. In this context, an example of a truncated word is *hou(se)...home*. The speaker started to say the word *house*, but stopped because (s)he meant to say *home*. The phone sequence associated with *hou* is thus a truncated word. An additional problem with truncated words is that the phone sequence can actually be an existing word as in *I...my*. A hesitation is usually associated with a filled pause, such as *uhm* or *mmm*. A big problem for an ASR system is a non-word which closely resembles an in-vocabulary word, e.g., in the previous example *ploem-bloem* (Eng: *flower*) or *fourf-fourth*: it is likely to be recognised as an in-vocabulary item. This problem is also to be expected in the case of a truncated word where the first phone sequence resembles an existing word.

In SpeM, the combination of the incremental recognition process, the Possible Word Constraint (PWC; Norris et al., 1997) implementation, and the garbage symbols makes it possible not only to detect out-of-vocabulary phone sequences (thus phone sequences that do not match an in-vocabulary word), but also to parse them and check whether they are either a possible word, which could in the future, in a subsequent pass, be included in the lexicon, or whether it is a phone sequence that cannot be a word.

The PWC is a lexical viability constraint that penalises a candidate word or word-initial cohort if a stretch of speech between the edge of the previous candidate word and the

location of a likely word boundary is itself not a possible word. In SpeM's implementation, a stretch of speech is not a possible word if it does not contain a vowel (of course, other methods to check the viability of a sequence of phones as a word are also possible). The PWC is implemented using 'garbage' phones. A garbage phone is effectively a phone that matches all other phones with the same cost. It is hypothesised whenever a phone insertion that is not word-internal occurs on a path. A (sequence of) garbage phone(s) that does not contain a vowel is marked as being a non-word, and the parse is penalised accordingly. Just like human listeners and unlike conventional ASR systems, SpeM is thus able to spot phone sequences – due, for instance, to hesitations or restarts – that are phonotactically incorrect and treat them as something that is a non-word. This can help with recognition and segmentation. Likewise, in SpeM, if the (sequence of) garbage phone(s) does contain a vowel, this might be a novel word. The current implementation of SpeM is not able to include this novel word in its lexicon, but it is possible to extend SpeM with this feature.

Thanks to the incremental search, SpeM is also able to do things that conventional ASR systems *cannot*. For instance, as already shown in Chapter 4, it allows SpeM to recognise words before their acoustic offset and, as described in Chapter 5, to recognise word-initial cohorts that can be used for creating utterance-specific lexicons. In short, SpeM is able to parse input that does not solely consist of sequences of lexical items.

#### *'Early' recognition*

In the case of recognising words before their acoustic offset (i.e., 'early' recognition), there are still a few issues left. During the recognition process of a word there are 'winners' at various stages: there is an interim winner, i.e., the word-initial cohort on the best path; there is a winner at the acoustic offset of the word; and there is a winner after the processing of the complete utterance. In Chapter 4, we searched for a (set of) predictor(s) for the correct recognition of a word after the processing of the complete utterance on the basis of the interim winner. The experiments described in Chapter 4 show that the predictors used to determine whether a word is correctly recognised before its acoustic offset are promising, but that many false accepts occur. In order to be able to use 'early' recognition in an actual system, a couple of problems need to be resolved. First of all, the performance of SpeM as a conventional ASR system needs to be improved. Second, the predictors that determine whether a word is indeed correctly recognised before the acoustic offset of the word is reached should be improved or better ones should be found. The number of false accepts will automatically be reduced if a solution has been found for the two problems.



### 6.1.3 One-stage vs. multi-stage recognition systems

As explained in Chapter 1, in the past most mainstream ASR systems were one-stage recognition systems, while lately more multi-stage recognition systems are being developed. In the introduction, two types of multi-stage recognition systems were distinguished. In the first type, the result of the first stage of the recognition system is used to tune the second stage recogniser, after which a full recognition attempt of the acoustic signal is carried out with the tuned system. In the second type, the first stage recogniser creates a segmental representation (often of phones) of the speech signal, after which a subsequent recognition step carries out a search through the phone graph.

The ASR system introduced in Chapter 5 is a multi-stage recognition system of the first type. On the basis of the stretches of speech marked as being out-of-vocabulary by the first recognition step, for each utterance a list of words is extracted from a much bigger lexicon. This list of words is then added to the lexicon of the second stage recogniser, yielding a lexicon that is more tuned to the utterance to be recognised. Subsequently, a new full recognition attempt is carried out. In this way, the lexicons at both the first and the second stage are kept small, and the number of out-of-vocabulary words is reduced in the second recognition step, both of which improve recognition performance as is shown in Chapter 5.

SpeM as developed in Chapter 3 is a multi-stage ASR system of the second type. In the first stage, the acoustic signal is transformed into a probabilistic phone graph; the second stage recognises words from the phone graph without using the acoustic signal again. SpeM's multi-stage approach is based on the cascaded processing of speech as is done by human listeners (see also Chapter 3).

### 6.1.4 Word activation vs. path-based scores

In Chapter 3, we proposed a method to convert the path-based scores that are used in ASR search methods, and thus also in SpeM, into the word-based activation scores used in models of human speech recognition. The details of the underlying mathematics were presented in Chapter 4.

The word activation of a word  $W$  is closely related to the probability  $P(W|X)$  of observing a word  $W$ , given the signal  $X$ . This can be rewritten using Bayes' Rule:

$$P(W | X) = \frac{P(X | W) \cdot P(W)}{P(X)}, \quad (6-1)$$

in which  $P(W)$  is the prior probability of  $W$ , and  $P(X)$  denotes the prior probability of observing the signal  $X$ . Bayes' Rule and the probability  $P(W|X)$  play a central role in the mathematical framework on which statistical pattern matching techniques are built (i.e., most ASR implementations). The Bayesian decomposition of the probability  $P(W|X)$  is the foundation on which we based the calculation of word activation.

The Bayesian activation, which is defined similar to  $P(W|X)$ , is calculated for the word and the path on which the word occurs. The Bayesian word activation is used as one of the predictors for early recognition in Chapter 4: for a word to be recognised, the word had to have a minimum Bayesian word activation, and the quotient of the activation of the word on the first-best path and its direct competitor had to exceed a pre-set threshold.

During human word recognition words are in competition, which means that words that overlap in the input actively inhibit each other. To ensure a fair competition, the word activation scores that are calculated by SpeM for the modelling of human word recognition should be comparable across paths and over time. To that end, in Chapter 3, the method for calculating word activation is extended. First, we computed the product of the Bayesian word activation and the Bayesian activation of the path it features on. Next, this score is divided by the total probability mass across all paths, yielding a word activation score that is normalised over paths and time.

The word activation as calculated by SpeM in Chapter 3 is, however, not based on 'active' inhibition (like the inhibition between lexical representations in the Shortlist model). It models competition between words in a 'static' way. The question remains whether the current way of modelling competition suffices or whether an active inhibition is necessary. In Chapter 3, three simulations were run and the results showed that SpeM was able to model correctly the outcomes of three psycholinguistic studies. It might be the case, however, that psycholinguistic studies exist or new results in the future will show that an active inhibition is necessary. In that case, the word activation calculation procedure should be refined and the issue that arises then is how this active inhibition should be implemented.

## 6.2 Future work

Although the results of SpeM are quite promising, there is room for improvement and expansion of the model. First of all, as already pointed out in Section 6.1.1, SpeM's performance – both as an ASR system as well as a computational model of HSR – might benefit from a better front end. A challenge for the future, thus, lies in building an APR that is able to create a probabilistic phone graph in which the 'correct' phonemes are present as often as possible and also have a high probability. As explained before, no top-down

information may be used. Two possible ways of improving the APR without using top-down information would be to build a language model for the APR that models the phonotactic constraints of the language better, or to use a different type of discrete symbolic intermediate representation of the acoustic signal, e.g., features instead of phones. A different type of solution to the issue of improving the performance of the APR would be to add pronunciation variants for each word in SpeM's lexicon, such that SpeM's search mechanism is better able to deal with paths through the phone graph in which phones are inserted or deleted in comparison with the canonical phonemic transcriptions of words.

The incremental search in SpeM makes it an ASR system that is able to do things that conventional ASR systems cannot. For instance, SpeM is able to spot out-of-vocabulary words and mark them as possible new words. In order to overcome the problem of recognition errors due to out-of-vocabulary words, these new words should be added to SpeM's lexicon. As already pointed out in Section 6.1.1, the current version of SpeM is not able to actually update its lexicon, but this feature could be added. However, the procedure to determine whether a phone sequence can be a word (the PWC) is rather coarse: a phone sequence is a possible word when it contains a vowel. Of course, when the phonotactic rules of the language are being violated, these phone sequences are not words. So, before the inclusion of the phone string as a new word in a lexicon, an additional procedure is needed to check whether the phone string is indeed a word. Dusan and Flanagan (2002), for instance, present a multi-model system that asks the user to provide additional information by pointing with a mouse on a computer screen when a word is encountered that is not part of the lexicon. This information is then used to add the word and its semantics to the system. A different approach would be to ask a user to spell the word that has been identified as out-of-vocabulary. Chung et al. (2003) describe an ASR system that is able to recognise spelled words and add them to the recogniser's lexicon and a natural language grammar.

Chapter 4 showed that the predictors used for early recognition are promising, but that there is room for improvement. To decrease the number of false accepts SpeM would benefit from improved or new predictors for early recognition. In the current implementation, the Bayesian word activation (thus in contrast to the word activation used for the experiments described in Chapter 3, the Bayesian word activation is not normalised over paths and time) is used as one of the predictors. The Bayesian word activation does not have the path score incorporated. Thus, one way to improve on this predictor might be to use the time- and path-normalised word activation score.

In Chapter 3, it was shown that SpeM is able to model findings from three different types of behavioural studies of human word recognition. These results suggest that SpeM is

indeed able to model substantial parts of the human word recognition process. There are, however, more aspects to human speech recognition than those associated with word recognition. For SpeM to be able to model all these aspects of the human speech recognition process, several issues remain to be resolved. For instance, SpeM is not able to model effects of phonemic context on speech recognition due to coarticulation and other connected speech phenomena, processes involved in phoneme recognition, and lexical effects on phonetic perception (see for a review, McQueen, 2004). To model these processes, SpeM has to be extended, for instance by adding a decision layer similar to the one implemented in Merge (Norris et al., 2000). Second, as research by Goldinger (1998) shows, human listeners are able to remember details of specific tokens of words that they have heard, and these memories for words have shown to influence subsequent speech processing. One way for SpeM to be able to model these results, is to adapt the APR module such that it is able to provide information about the speaker as well. A simple first step would be to train gender-dependent phone models. Finally, the current version of SpeM is able to use unigram and bigram language models. Humans, however, are able to use more contextual information than just the word frequency and/or the probability of co-occurrence of the current and the previous word (e.g., Marslen-Wilson, 1987). Experiments by, e.g., Zwitserlood (1989) have shown that context information is used after lexical access. For SpeM to be able to simulate these results, higher-order language models such as long-span language models and grammars should be included. The inclusion of higher-order language models will also benefit the performance of SpeM as a conventional ASR system.

### 6.3 Concluding remarks

This thesis started with two citations that inspired my research project. The goal of this research was to narrow the gap that has existed between the research fields of human and automatic speech recognition:

*“The central issues in the study of speech recognition by human listeners (HSR) and of automatic speech recognition (ASR) are [...] clearly comparable; nevertheless, the research communities that concern themselves with ASR and HSR are largely distinct.”*

- R. K. Moore & A. Cutler (2001)

The fields of human and automatic speech recognition both study the speech recognition process. This suggests that there are close parallels between the two research fields, which

were made explicit by a computational-level analysis (Marr, 1982) of the word recognition process.

The second citation provided the starting point for the endeavour of narrowing the gap between HSR and ASR:

*“Given the relatively advanced state of psycholinguistics and speech perception, it seems remarkable that the only working models of lexical access from acoustic waveforms are products of the engineering technology of automatic speech recognition [...].”*

- T. M. Nearey (2001)

I followed Nearey’s (2001) suggestion of combining dynamic pattern recognition techniques from ASR with computational models of HSR in order to be able to use “detailed phonetic models [...] as front ends for reasonable models of lexical access”, although he doubted that existing HSR models “will work as advertised when attached to real phonetic transduction systems”. In this thesis, I presented the end-to-end computational model of human word recognition, SpeM, built using techniques from ASR. SpeM has proven to be successful in simulating parts of the human word recognition process. Nearey was right that coupling an existing HSR model, viz. Shortlist, and an APR did not work. But SpeM has shown that techniques and knowledge from ASR can be used to build a “working model of lexical access from acoustic waveforms”.

The close parallels between the two research fields were further revealed by the development of the word activation scores used by SpeM. I proved that it is possible to calculate a word-based continuous activation score from path-based ASR scores that is comparable to the activation scores used in HSR. The results showed that a left-to-right path-based decoding strategy as used in ASR systems (and in SpeM) is able to model the word-based competition effects found in behavioural studies of human speech recognition.

In conclusion, the most obvious contribution that ASR can make to HSR is to assist in the development of models that are able to account for the complete human speech recognition process from the acoustic analysis to the recognition of words in continuous speech. SpeM has proven to be a successful first step in that direction. The contribution of HSR to ASR is not yet as clear-cut. However, if a model of the complete human speech recognition process would exist, it could lead to interesting new ideas for the development of better ASR systems.



## Bibliography

- Alloppenna, P.D., Magnuson, J.S., Tanenhaus, M.K. (1998). Tracking the time course of spoken word recognition using eye movements: Evidence for continuous mapping models. *Journal of Memory and Language*, 38, 419-439.
- Anderson, J.R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.
- Andruski, J.E., Blumstein, S.E., Burton, M. (1994). The effect of subphonetic differences on lexical access. *Cognition*, 52, 163-187.
- Ball, M.J., Rahilly, J. (2002). Transcribing disordered speech: The segmental and prosodic layers. *Clinical Linguistics & Phonetics*, 16 (5), 329-344.
- Bard, E.G., Shillcock, R.C., Altmann, G.T.M. (1988). The recognition of words after their acoustic offsets in spontaneous speech: Effects of subsequent context. *Perception and Psychophysics*, 44(5), 395-408.
- Bazzi, I., Glass, J.R. (2000). Modeling out-of-vocabulary words for robust speech recognition. *Proceedings of ICSLP*, Beijing, China, pp. 401-404.
- Bazzi, I., Glass, J.R. (2001). Learning units for domain-independent out-of-vocabulary word modeling. *Proceedings of Eurospeech*, Aalborg, Denmark, pp. 61-64.
- Bouwman, G., Boves, L., Koolwaaij, J. (2000). Weighting phone confidence measures for automatic speech recognition. *Proceedings of the COST249 Workshop on Voice Operated Telecom Services*, Ghent, Belgium, pp. 59-62.
- Brakensiek, A., Rottland, J., Rigoll, G. (2003). Confidence measures for an address reading system. *Proceedings of the IEEE International Conference on Document Analysis and Recognition*, cdrom.
- Burnage, G. (1990). *CELEX: A guide for users*. Nijmegen, The Netherlands: CELEX.
- Cardillo, P.S., Clements, M., Miller, M.S. (2002). Phonetic searching vs. LVCSR: How to find what you really want in audio archives. In *International Journal of Speech Technology*, 5(1) (pp. 9-22). The Netherlands: Springer Science+Business Media B.V.
- Carpenter, B. (1999). Human versus machine: Psycholinguistics meets ASR. *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Keystone, CO, pp. 225-228.
- Chung, G., Wang, C., Seneff, S., Filisko, E., Tang, M. (2004). Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation. *Proceedings of Interspeech*, Jeju Island, Korea, pp. 328-332.
- Chung, G., Seneff, S., Wang, C. (2003). Automatic acquisition of names using speak and spell mode in spoken dialogue systems. *Proceedings of HLT-NAACL*, Edmonton, Canada, pp. 197-200.
- Chung, G., Seneff, S., Wang, C., Hetherington, I.L. (2004). A dynamic vocabulary spoken dialogue interface. *Proceedings of Interspeech*, Jeju Island, Korea, pp. 327-330.
- Church, K. (1987). Phonological parsing and lexical retrieval. *Cognition*, 25, 53-69.

- Cluff, M.S., Luce, P.A. (1990). Similarity neighborhoods of spoken two-syllable words: Retroactive effects on multiple activation. *Journal of Experimental Psychology: Human Perception and Performance*, 16, 551-563.
- Cucchiaroni, C. (1993). Phonetic transcription: A methodological and empirical study. *Ph.D. thesis*, University of Nijmegen, The Netherlands.
- Cucchiaroni, C., Binnenpoorte, D., Goddijn, S. (2001). Phonetic transcriptions in the Spoken Dutch Corpus: How to combine efficiency and good transcription quality. *Proceedings of Eurospeech*, Aalborg, Denmark, pp. 1679-1682.
- Cutler, A. (1998). The recognition of spoken words with variable representations. *Proceedings of the ESCA Workshop on Sound Patterns of Spontaneous Speech*, Aix-en-Provence, France, pp. 83-92.
- Cutler, A., Demuth, K., McQueen, J.M. (2002). Universality versus language-specificity in listening to running speech. *Psychological Science*, 13, 258-262.
- Cutler, A., Norris, D. (1988). The role of strong syllables in segmentation for lexical access. *Journal of Experimental Psychology: Human Perception and Performance*, 14, 113-121.
- Davis, M.H., Marslen-Wilson, W.D., Gaskell, M.G. (2002). Leading up the lexical garden-path: Segmentation and ambiguity in spoken word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 28, 218-244.
- Demuyne, K., Laureys, T., Van Compernelle, D., Van hamme, H. (2003). FlaVoR: A flexible architecture for LVCSR. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 1973-1976.
- Dusan, S., Flanagan, J. (2002). Adaptive dialog based upon multimodal language acquisition. *Proceedings of the IEEE International Conference on Multimodal Interfaces*, Pittsburgh, PA, pp. 135-140.
- Elman, J.L., McClelland, J.L. (1986). Exploiting lawful variability in the speech wave. In J.S. Perkell & D.H. Klatt (Eds.), *Invariance and variability of speech processes* (pp. 360-380). Hillsdale, NJ: Erlbaum.
- Frauenfelder, U.H., Kearns, R.K. (1996). Sequence Monitoring. *Language and Cognitive Processes*, 11, 665-673.
- Furui, S. (1996). An overview of speaker recognition technology. In C.-H. Lee, F.K. Soong & K.K. Paliwal (Eds.), *Automatic Speech and Speaker Technology* (pp. 31-56). Boston: Kluwer Academic Publishers.
- Gaskell, M.G., Dumay, N. (2003). Lexical competition and the acquisition of novel words. *Cognition*, 89, 105-132.
- Gaskell, M.G., Marslen-Wilson, W.D. (1997). Integrating form and meaning: A distributed model of speech perception. *Language and Cognitive Processes*, 12, 613-656.
- Gaskell, M.G., Marslen-Wilson, W.D. (1998). Mechanisms of phonological inference in speech perception. *Journal of Experimental Psychology: Human Perception and Performance*, 24, 380-396.



- Geutner, P., Finke, M., Waibel, A. (1999). Selection criteria for hypothesis driven lexical adaptation. *Proceedings of ICASSP*, Phoenix, AZ, pp. 617-620.
- Gauvain, J.L., Lamel, L.F., Adda, G., Adda-Decker, M. (1994). Speaker-independent continuous speech dictation. *Speech Communication*, 15 (1), 21-27.
- Glass, J.R. (2003). A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17, 137-152.
- Glass, J.R., Hazen, T.J., Hetherington, I.L. (1999). Real-time telephone-based speech recognition in the Jupiter domain. *Proceedings of ICASSP*, Phoenix, AZ, pp. 61-64.
- Goldinger, S. (1996). Auditory Lexical Decision. *Language and Cognitive Processes*, 11, 559-567.
- Goldinger, S.D. (1998). Echoes of echoes?: An episodic theory of lexical access. *Psychological Review*, 105, 251-279.
- Gow, D.W., Jr. (2002). Does English coronal place assimilation create lexical ambiguity? *Journal of Experimental Psychology: Human Perception and Performance*, 28, 163-179.
- Gow, D.W., Gordon, P.C. (1995). Lexical and prelexical influences on word segmentation: Evidence from priming. *Journal of Experimental Psychology: Human Perception and Performance*, 21, 344-359.
- Grosjean, F. (1985). The recognition of words after their acoustic offset: Evidence and implications. *Perception & Psychophysics*, 28, 299-310.
- Grosjean, F. (1996). Gating. *Language and Cognitive Processes*, 11, 597-604.
- Hawkins, S. (2003). Roles and representations of systematic fine phonetic detail in speech understanding. *Journal of Phonetics*, 31, 373-405.
- Hazen, T.J. (2000). A comparison of novel techniques for rapid speaker adaptation. *Speech Communication*, 31, 15-33.
- Hetherington, L. (1994). A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding, *Ph.D. Thesis*, MIT, Cambridge, MA.
- Hetherington, L. (1995). New words: Effect on recognition performance and incorporation issues. *Proceedings of Eurospeech*, Madrid, Spain, pp. 1645-1648.
- Hermansky, H. (2001). Human speech perception: Some lessons from automatic speech recognition. In R. Smits, J. Kingston, T.M. Nearey & R. Zondervan (Eds.), *Proceedings of the workshop on speech recognition as pattern classification* (pp. 61-66). Nijmegen, MPI for Psycholinguistics.
- Hintzman, D.L. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological Review*, 93, 411-428.
- Hirose, K., Minematsu, N., Hashimoto, Y., Iwano, K. (2001). Continuous speech recognition of Japanese using prosodic word boundaries detected by mora transition modeling of fundamental frequency contours. *Proceedings of the Workshop on*

- Prosody in Automatic Speech Recognition and Understanding*, Red Bank, NJ, pp. 61-66.
- Höge, H., Draxler, C., van den Heuvel, H., Johansen, F.T., Sanders E., Tropf, H.S. (1999). Speechdat multilingual speech databases for teleservices: Across the finish line. *Proceedings of Eurospeech, Budapest, Hungary*, pp. 2699-2702.
- Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, 359-366.
- Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge, MA.: MIT Press.
- Juang, B.H., Furui, S. (Eds., 2000). Spoken language processing. *Special Issue of the Proceedings of the IEEE*, 88 (8).
- Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, 20, 137-194.
- Kessens, J.M., Wester, M., Strik, H. (1999). Improving the performance of a Dutch CSR by modelling within-word and cross-word pronunciation variation. *Speech Communication*, 29, 193-207.
- Kessens, J.M., Cucchiaroni, C., Strik, H. (2003). A data-driven method for modeling pronunciation variation. *Speech Communication*, 40, 517-534.
- Klatt, D.H. (1977). Review of the ARPA speech understanding project. *Journal of the Acoustical Society of America*, 62 (6), 1345-1366.
- Klatt, D.H. (1979). Speech perception: A model of acoustic-phonetic analysis and lexical access. *Journal of Phonetics*, 7, 279-312.
- Klatt, D.H. (1989). Review of selected models of speech perception. In W.D. Marslen-Wilson (Ed.), *Lexical representation and process* (pp. 169-226). Cambridge, MA: MIT Press.
- Laureys, T., Vandeghinste, V., Duchateau, J. (2002). A hybrid approach to compounds in LVCSR. *Proceedings of ICSLP*, Denver, CO, pp. 697-700.
- Lesser, V.R., Fennell, R.D., Erman, L.D., Reddy, D.R. (1975). Organisation of the hearsay – II: Speech understanding system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 11-23.
- Lippmann, R. (1997). Speech recognition by machines and humans. *Speech Communication*, 22 (1), 1-15.
- Luce, P.A. (1986). A computational analysis of uniqueness points in auditory word recognition. *Perception & Psychophysics*, 39, 155-158.
- Luce, P.A., Goldinger, S.D., Auer, E.T., Vitevitch, M.S. (2000). Phonetic priming, neighborhood activation, and PARSYN. *Perception & Psychophysics*, 62, 615-625.
- Luce, P.A., Pisoni, D.B. (1998). Recognising spoken words: The neighborhood activation model. *Ear and Hearing*, 19, 1-36.
- Maddieson, I. (1984). *Patterns of sounds*. Cambridge: Cambridge University Press.

- Magnuson, J.S., McMurray, B., Tanenhaus, M.K., Aslin, R.N. (2003). Lexical effects on compensation for coarticulation: The ghost of Christmash past. *Cognitive Science*, 27, 285-298.
- Makhoul, J., Kubala, F., Leek, T., Liu, D., Nguyen, L., Schwartz, R., Srivastava, A. (2000). Speech and language technologies for audio indexing and retrieval. *Special Issue of the Proceedings of the IEEE*, 88 (8), pp. 1338-1353.
- Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: Freeman & Co.
- Marslen-Wilson, W.D. (1987). Functional parallelism in spoken word recognition. *Cognition*, 25, 71-102.
- Marslen-Wilson, W.D. (1990). Activation, competition, and frequency in lexical access. In G.T.M. Altmann (Ed.), *Cognitive models of speech processing: Psycholinguistic and computational perspectives* (pp. 148-172). Cambridge, MA: MIT Press.
- Marslen-Wilson, W.D., Tyler, L. (1980). The temporal structure of spoken language understanding. *Cognition*, 8, 1-71.
- Marslen-Wilson, W.D., Warren, P. (1994). Levels of perceptual representation and process in lexical access: Words, phonemes, and features. *Psychological Review*, 101, 653-675.
- Marslen-Wilson, W.D., Welsh, A. (1978). Processing interactions and lexical access during word-recognition in continuous speech. *Cognitive Psychology*, 10, 29-63.
- McClelland, J.L., Elman, J.L. (1986). The TRACE model of speech perception. *Cognitive Psychology*, 18, 1-86.
- McQueen, J.M. (1996). Phonetic Categorisation. *Language and Cognitive Processes*, 11, 655-664.
- McQueen, J.M. (1998). Segmentation of continuous speech using phonotactics. *Journal of Memory and Language*, 39, 21-46.
- McQueen, J.M. (2003). The ghost of Christmas future: Didn't Scrooge learn to be good? Commentary on Magnuson, McMurray, Tanenhaus and Aslin (2003). *Cognitive Science*, 27, 795-799.
- McQueen, J.M. (2004). Speech perception. In K. Lamberts & R. Goldstone (Eds.), *The handbook of cognition* (pp. 255-275). London: Sage Publications.
- McQueen, J.M., Cutler, A., Briscoe, T., Norris, D. (1995). Models of continuous speech recognition and the contents of the vocabulary. *Language and Cognitive Processes*, 10, 309-331.
- McQueen, J.M., Cutler, A., Norris, D. (in preparation). Decoding speech via abstract representations of speech sounds: Evidence from perceptual learning.
- McQueen, J.M., Dahan, D., Cutler, A. (2003). Continuity and gradedness in speech processing. In A.S. Meyer & N.O. Schiller (Eds.), *Phonetics and phonology in language comprehension and production: Differences and similarities* (pp. 39-78). Berlin: Mouton de Gruyter.

- McQueen, J.M., Norris, D., Cutler, A. (1994). Competition in spoken word recognition: Spotting words in other words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20, 621-638.
- McQueen, J.M., Norris, D., Cutler, A. (1999). Lexical influence in phonetic decision-making: Evidence from subcategorical mismatches. *Journal of Experimental Psychology: Human Perception and Performance*, 25, 1363-1389.
- Meddis, R., Hewitt, M.J. (1991). Virtual pitch and phase sensitivity of a computer model of the auditory periphery: I. Pitch identification. *Journal of the Acoustical Society of America*, 89, 2866-2882.
- Moore, R.K. (2003). A comparison of the data requirements of automatic speech recognition systems and human listeners. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 2581-2584.
- Moore, R.K., Cutler, A. (2001). Constraints on theories of human vs. machine recognition of speech. In R. Smits, J. Kingston, T.M. Nearey & R. Zondervan (Eds.), *Proceedings of the workshop on speech recognition as pattern classification* (pp. 145-150). Nijmegen, MPI for Psycholinguistics.
- Nearey, T.M. (2001). Towards modelling the perception of variable-length phonetic strings. In R. Smits, J. Kingston, T.M. Nearey & R. Zondervan (Eds.), *Proceedings of the workshop on speech recognition as pattern classification* (pp. 133-138). Nijmegen, MPI for Psycholinguistics.
- Ney, H., Aubert, X. (1996). Dynamic programming search: From digit strings to large vocabulary word graphs. In C.-H. Lee, F.K. Soong, & K.K. Paliwal (Eds.), *Automatic Speech and Speaker Recognition* (pp. 385-413). Boston: Kluwer Academic Publishers.
- Ney, H., Ortmanns, S. (2000). Progress in dynamic programming search for LVCSR. *Proceedings of the IEEE*, 88 (8), 1224-1240.
- Norris, D. (1982). Autonomous processes in comprehension: A reply to Marslen-Wilson and Tyler. *Cognition*, 11, 97-101.
- Norris, D. (1986). Word recognition: Context effects without priming. *Cognition*, 22, 93-136.
- Norris, D. (1994). Shortlist: A connectionist model of continuous speech recognition. *Cognition*, 52, 189-234.
- Norris, D. (in press). How do computational models help us develop better theories? In A. Cutler (Ed.), *Twenty-first century psycholinguistics: Four cornerstones*. Hillsdale, NJ: Erlbaum.
- Norris, D., McQueen, J.M., Cutler, A. (1995). Competition and segmentation in spoken-word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21, 1209-1228.
- Norris, D., McQueen, J.M., Cutler, A. (2000). Merging information in speech recognition: Feedback is never necessary. *Behavioural and Brain Sciences*, 23, 299-325.

- Norris, D., McQueen, J.M., Cutler, A. (2003). Perceptual learning in speech. *Cognitive Psychology*, 47, 204-238
- Norris, D., McQueen, J.M., Cutler, A., Butterfield, S. (1997). The possible-word constraint in the segmentation of continuous speech. *Cognitive Psychology*, 34, 191-243.
- Norris, D., McQueen, J.M., Smits, R. (in preparation). Shortlist II: A Bayesian model of continuous speech recognition.
- Ohtsuki, K., Hiroshima, N., Matsunaga, S., Hayashi, Y. (2004). Multi-pass ASR using vocabulary expansion. *Proceedings of Interspeech*, Jeju Island, Korea, pp. 1713-1716.
- Patterson, R.D., Allerhand, M., Giguere, C. (1995). Time-domain modelling of peripheral auditory processing: A modular architecture and a software platform. *Journal of the Acoustical Society of America*, 98, 1890-1894.
- Paul, D.B. (1992). An efficient A\* stack decoder algorithm for continuous speech recognition with a stochastic language model. *Proceedings of ICASSP*, San Francisco, CA, pp. 25-28.
- Perkell, J.S., Klatt, D.H. (Eds., 1986) *Invariance and variability of speech processes*. Hillsdale, NJ: Lawrence Erlbaum Associates
- Pitt, M.A., McQueen, J.M. (1998). Is compensation for coarticulation mediated by the lexicon? *Journal of Memory and Language*, 39, 347-370.
- Rabiner, L., Juang, B.-H. (1993). *Fundamentals of speech processing*. New Jersey: Prentice Hall.
- Radeau, M., Morais, J., Mousty, P., Bertelson, P. (2000). The effect of speaking rate on the role of the uniqueness point in spoken word recognition. *Journal of Memory and Language*, 42 (3), 406-422.
- Salverda, A.P., Dahan, D., McQueen, J.M. (2003). The role of prosodic boundaries in the resolution of lexical embedding in speech comprehension. *Cognition*, 90, 51-89.
- Samuel, A.G. (2001). Knowing a word affects the fundamental perception of the sounds within it. *Psychological Science*, 12, 348-351.
- Samuel, A.G., Pitt, M.A. (2003). Lexical activation (and other factors) can mediate compensation for coarticulation. *Journal of Memory and Language*, 48, 416-434.
- Scharenborg, O., Boves, L. (2002a). Pronunciation variation modelling in a model of human word recognition. *Proceedings of the Workshop on Pronunciation Modelling and Lexicon Adaptation*, Estes Park, CO, pp. 65-70.
- Scharenborg, O., Boves L., de Veth, J. (2002b). ASR in a human word recognition model: generating phonemic input for Shortlist. *Proceedings of ICSLP*, Denver, CO, pp. 633-636.
- Scharenborg, O., Boves, L., ten Bosch, L. (2004). 'On-line early recognition' of polysyllabic words in continuous speech. *Proceedings of the Tenth Australian International Conference on Speech Science & Technology*, Sydney, Australia, cdrom.

- Scharenborg, O., McQueen, J.M., ten Bosch, L., Norris, D. (2003a). Modelling human speech recognition using automatic speech recognition paradigms in SpeM. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 2097-2100.
- Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M. How should a speech recognizer work? *Accepted for publication in Cognitive Science*.
- Scharenborg O., ten Bosch, L., Boves, L. (2003b). Recognising 'real-life' speech with SpeM: A speech-based computational model of human speech recognition. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 2285-2288.
- Scharenborg O., ten Bosch, L., Boves, L. (2003c). 'Early recognition' of words in continuous speech. *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, St. Thomas, US Virgin Islands, cdrom.
- Scharenborg, O., ten Bosch, L., Boves, L., Norris, D. (2003d). Bridging automatic speech recognition and psycholinguistics: Extending Shortlist to an end-to-end model of human speech recognition. *Journal of the Acoustical Society of America*, 114 (6), 3032-3035.
- Seneff, S. (2002). Response planning and generation in the Mercury flight reservation system. *Computer Speech and Language*, 16, 283-312.
- Seneff, S. (2004). The use of subword linguistic modeling for multiple tasks in speech recognition. *Speech Communication*, 42 (3-4), 373-390.
- Seneff, S., Wang, C., Hazen, T.J. (2003). Automatic induction of N-gram language models from a natural language grammar. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 641-644.
- Shriberg, L.D., Kwiatkowski, J., Hoffmann, K. (1984). A procedure for phonetic transcription by consensus. *Journal of Speech and Hearing Research*, 27, 456-465.
- Spinelli, E., McQueen, J.M., Cutler, A. (2003). Processing resyllabified words in French. *Journal of Memory and Language*, 48, 233-254.
- Steinbiss, V., Ney, H., Haeb-Umbach, R., Tran, B.-H., Essen, U., Kneser, R., Oerder, M., Meier, H.-G., Aubert, X., Dugast, C., Geller, D. (1993). The Philips research system for large-vocabulary continuous speech recognition. *Proceedings of Eurospeech*, Berlin, Germany. pp. 2125-2128.
- Stevens, K.N. (2002). Toward a model for lexical access based on acoustic landmarks and distinctive features. *Journal of the Acoustical Society of America*, 111, 1872-1891.
- Stolcke, A., Shriberg, E., Tür, D., Tür, G. (1999). Modeling the prosody of hidden events for improved word recognition. *Proceedings of Eurospeech*, Budapest, Hungary, pp. 311-314.
- Strik, H., Russel, A.J.M., van den Heuvel, H., Cucchiaroni, C., Boves, L. (1997). A spoken dialog system for the Dutch public transport information service. *International Journal of Speech Technology*, 2 (2), 119-129.

- Sturm, J., Kamperman, H., Boves, L., den Os, E. (2000). Impact of speaking style and speaking task on acoustic models. *Proceedings of ICSLP*, Beijing, China, pp. 361-364.
- Tabossi, P., Burani, C., Scott, D. (1995). Word identification in fluent speech. *Journal of Memory and Language*, 34, 440-467.
- Tabossi, P., Collina, S., Mazzetti, M., Zoppello, M. (2000). Syllables in the processing of spoken Italian. *Journal of Experimental Psychology: Human Perception and Performance*, 26, 758-775.
- Tang, M., Seneff, S., Zue, V. (2003). Two-stage speech recognition using feature-based models: a preliminary study. *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, St. Thomas, US Virgin Islands, cdrom.
- ten Bosch, L. (2001). ASR-HSR from an ASR point of view. In R. Smits, J. Kingston, T.M. Nearey & R. Zondervan (Eds.), *Proceedings of the workshop on speech recognition as pattern classification* (pp. 49-54). Nijmegen, MPI for Psycholinguistics.
- Vitevitch, M.S., Luce, P.A. (1998). When words compete: Levels of processing in spoken word recognition. *Psychological Science*, 9, 325-329.
- Vitevitch, M.S., Luce, P.A. (1999). Probabilistic phonotactics and neighborhood activation in spoken word recognition. *Journal of Memory and Language*, 40, 374-408.
- Vroomen, J., de Gelder, B. (1995). Metrical segmentation and lexical inhibition in spoken word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 21, 98-108.
- Waibel, A., Geutner, P., Mayfield Tomokiyo, L., Schultz, A., Woszczyna, M. (2000). Multilinguality in speech and spoken language systems. *Special Issue of the Proceedings of the IEEE*, 88 (8), pp. 1297-1313.
- Wessel, F., Schlueter, R., Macherey, K., Ney, H. (2001). Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9 (3), 288-298.
- Wester, M. (2003). Pronunciation modeling for ASR - knowledge-based and data-derived methods. *Computer Speech and Language*, 17 (1), 69-85.
- Woodland, P.C. (2001). Speaker adaptation for continuous density HMMs: A review. *Proceedings of the ISCA workshop on adaptation methods for speech recognition*, Sophia-Antipolis, France, pp.11-19.
- Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P. (2002). *The HTK book (for HTK version 3.2)*. Technical Report, Cambridge University, Engineering Department.
- Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L. (2000). Jupiter: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8 (1), 85-96.
- Zwitserslood, P. (1989). The locus of the effects of sentential-semantic context in spoken-word processing. *Cognition*, 32, 25-64.





## Summary

In everyday life, speech is all around us, on the radio, television, and in human-human interaction. Communication using speech is easy. We human beings are continually confronted with novel utterances that speakers select from the infinite set of possible utterances in a language, and usually we encounter little to no difficulty in recognising and understanding them. There are various research fields that investigate (parts of) the speech recognition process. In this thesis, I focus on two: the fields of *human* speech recognition (HSR) and of *automatic* speech recognition (ASR).

Although the two research areas are closely related – they both study the speech recognition process, and the central issue of both is word recognition – their aims are different. In HSR research, the goal is to understand how we, as listeners, recognise spoken utterances. This is done by building models that can be used for the simulation and explanation of the human speech recognition *process*. In ASR, the central issue is minimising the number of recognition errors, irrespective of the question whether the approach parallels the processes used by humans. In parallel with the difference in aims between the two research fields, the research approaches are different as well.

The most important difference between the two fields for this thesis is that although both ASR and HSR claim to investigate the whole recognition process from the acoustic signal to the recognised units, an automatic speech recogniser necessarily is an end-to-end system, while most models of HSR describe only parts of the human speech recognition process. An integral model covering all stages of the human speech recognition process does not yet exist. One part of the recognition process that virtually all models of HSR lack is a module that converts the acoustic signal into some kind of segmental representation. Most existing HSR models cannot recognise real speech. This makes it hard to evaluate the theoretical assumptions underlying models of HSR in real-life test conditions.

Despite the gap that separates ASR and HSR, there is a growing interest in possible cross-fertilisation. The central goal of this thesis is to narrow this gap that has existed for decades between the two research fields. This endeavour is started from the field of HSR by trying to build an end-to-end model of human word recognition using techniques from the field of ASR. An end-to-end model is able to simulate the speech recognition process from the acoustic signal to the recognised words.

### **Human speech recognition**

To investigate the properties underlying the human speech recognition process, HSR experiments with human subjects are usually carried out in a laboratory environment. In these experiments, various measurements are taken, such as reaction time and phoneme response probabilities. Based on these measurements, theories about specific parts of the human speech recognition system are developed or refined. To put the theories to further

test, they are implemented in the form of computational models for the simulation and explanation of HSR. For this thesis, the computational model of human word recognition *Shortlist* is of interest.

Symbolic theories of human speech recognition say that human listeners first map the incoming acoustic signal onto prelexical representations, e.g., in the form of phonemes, after which the prelexical representations are mapped onto the lexical representations, almost invariably in the form of words. The speech recognition process in symbolic theories thus consists of two levels: the prelexical level and the lexical level. A central requirement of symbolic computational models is thus a segmental representation of the speech signal. Since most computational models of HSR are not able to recognise real speech, they use a handcrafted ‘error-free’ linear representation of the input – in the sense that the input always perfectly aligns with the segmental representations of the words in the lexicon. Thus in effect, in most symbolic computational models, the process of creating the prelexical representations is only assumed, and not physically present. Only the output of the prelexical process is available in the form of the handcrafted segmental representation of the speech signal.

### **Automatic speech recognition**

The input of an ASR system consists of an acoustic signal. During speech recognition, the speech signal is first passed through the acoustic pre-processor where feature vectors are extracted from the speech signal. Subsequently, the feature vectors are matched with the succession of acoustic models associated with the items, usually words, in the internal lexicon. For each feature vector the degree of fit between the feature vector and each of the models is determined. Ultimately, the word that belongs to the sequence of acoustic models for which the degree of fit with the feature vectors in the input is best is hypothesised. An ASR system can only recognise those words that are included in its lexicon. Unless the system comes with a ‘reject’ option, an unknown word in the input will always be recognised as one of the words in its lexicon; this will result in an incorrect recognition.

For recognition, most ASR systems use an *integrated search*: all information (from the acoustic model set, lexicon, and language model) is used at the same time. Together these information sources form the search space. Speech recognition, then, is finding the best path through the search space. The likelihood of a number of hypothesised word sequences (paths) through the complete graph is computed, and then a trace back is performed to identify the words that were recognised on the basis of the hypothesis with the highest score at the end of the utterance. ASR systems are usually evaluated in terms of accuracy, the percentage of the input words that is recognised correctly, or in terms of word error rate (WER), the number of inserted, deleted, and substituted words divided by the total number of words in the input.

ASR systems do not adhere to a theory of HSR like computational models of human speech recognition do. Thus, a standard ASR system cannot be used as a computational end-to-end model of human speech recognition. But an ASR system is able to make a segmental representation of the speech signal. In this thesis, an adapted ASR system, an automatic phone recogniser (APR), is used as the missing front-end that converts the acoustic signal into a symbolic representation that can be used in current HSR models. An APR functions the same as an ASR system with the exception that the lexicon consists of phones instead of words. Consequently, the output of an APR consists of (sequences or graphs) of phones instead of words.

### **The structure of this thesis**

In addition to the introductory chapter which is summarised above, this thesis consists of four articles (chapters 2 through 5) and a concluding chapter in which the findings of the research presented in Chapters 2 through 5 are discussed and put into perspective. Also, the main conclusions are drawn and suggestions for further research are presented.

The research presented in Chapters 2 and 3 shows the benefit that can be obtained by using techniques from the field of ASR for building models of HSR. The experiments described in Chapters 4 and 5 show the benefit for ASR of a recognition procedure that makes use of key aspects of the human speech recognition process. Below, of each chapter a summary of the experiments, results, and conclusions is presented.

### **Chapter 2: Extending Shortlist to an end-to-end model of human speech recognition**

In Chapter 2, a first attempt was made to create an end-to-end model of human word recognition using techniques from the field of ASR. To that end, Shortlist was extended with an acoustic front-end in the form of an APR, which created a segmental representation of the speech signal in the form of a phoneme string that was used as input for Shortlist. To test how well this joint model was able to actually recognise real speech, a recognition experiment is carried out. 10,510 utterances consisting of either a Dutch city name or ‘ik weet het niet’ (‘I don’t know’) spoken in isolation were presented to the joint model.

The experiments described in Chapter 2 illustrate the consequences of some of the simplifying assumptions made in Shortlist and other HSR models, and show the extent to which these assumptions need to be revised to produce end-to-end HSR models that are able to deal with real-speech input. The biggest shortcoming of the joint model of the APR and Shortlist is that it makes ‘hard’ decisions at the level of input phones. Shortlist requires a single string of phone symbols as input. This implies that the APR is forced to make ‘hard’ decisions about the segmental representation of the speech signal based only on the acoustic information. The second shortcoming of Shortlist is that the search in the Shortlist module of the joint model is a simple lexical look-up: a phone insertion or deletion will cause a misalignment of all subsequent phones with the words in the lexicon. The

experiments clearly show that a straightforward combination of the APR and Shortlist does not yield an end-to-end model of HSR that can deal satisfactorily with real-life input, even though Shortlist is a successful model for a specific aspect of the human speech recognition process.

### Chapter 3: How should a speech recogniser work?

As observed before, there is little communication between the research fields of ASR and HSR. In Chapter 3 it is suggested that one barrier to communication might be that the research is often seen as being about *how* humans, or *how* machines, recognise speech (according to Marr these questions are addressing the algorithmic and implementational levels of an information-processing system) instead of addressing the question *why* certain functions are needed for recognising speech (the computational level according to Marr). Chapter 3 describes a computational-level analysis of the word recognition process which made the close parallels between HSR and ASR explicit. The computational parallels were further illustrated by the development of SpeM (Speech-based model of HSR): a computational model of HSR, based on the theory underlying Shortlist, that was built using techniques from ASR. SpeM is not just a re-implementation of Shortlist; it represents an important advancement over existing models of HSR in that it is able to recognise words from acoustic speech input at reasonably high levels of accuracy, while currently existing models of HSR almost invariably assume a (error-free) symbolic representation of the acoustic signal as input.

In SpeM, the ‘hard decisions problem’ at the input level the joint model presented in Chapter 2 suffered from was solved by representing the speech signal as a probabilistic phone lattice containing multiple phone-string hypotheses instead of a one-dimensional phone string. This allows, in a natural way, the postponement of a hard decision to a point later in the word search process. The second shortcoming of the combination of the APR and Shortlist, the implementation of the lexical search, is solved in SpeM by using a search algorithm based on dynamic programming techniques that tolerates misalignments between the input and canonical phonemic lexical representations (at a certain cost).

Two types of experiments carried out with SpeM are presented in Chapter 3. The first experiment is identical to the recognition experiment with the joint model of Chapter 2. The results of this experiment show that SpeM strongly outperforms Shortlist in its ability to recognise words from real-life speech, spoken by a large number of different talkers in the type of acoustic environments found in normal life, largely due to the phone-lattice representation of the input in SpeM.

In the second type of experiment, SpeM’s computational power with respect to the simulation of the human speech recognition process is addressed. For these simulations, *word* activations are necessary, but SpeM calculates *path*-based scores. In Chapter 3, a method is presented that converts the path-based scores of SpeM into word-based

activation scores. Three simulations carried out with SpeM show that the model, with the acoustic signal as input, is able to simulate the same aspects of the human speech recognition process as Shortlist, which has a phoneme string as input.

#### Chapter 4: ‘Early recognition’ of polysyllabic words in continuous speech

Humans are well able to identify and recognise a word before its acoustic realisation is complete. This in contrast to conventional ASR systems, which compute the likelihood of a number of hypothesised word sequences, and identify the words that were recognised on the basis of the hypothesis with the highest score at the end of the utterance to maximise performance. Furthermore, in contrast to conventional integrated search methods used in ASR systems, SpeM uses an incremental search. This incremental search gives a ranked list of hypotheses at each moment in time *during* the speech recognition process and is therefore able to recognise a word *before* its acoustic offset. Chapter 4 investigates how SpeM’s incremental search can be used for the recognition of a word before its acoustic realisation is complete; this is referred to as ‘early recognition’.

Experiments on 1,463 polysyllabic ‘focus’ words in 885 utterances showed that 64.0% (936 utterances) were recognised correctly at the end of the utterance. For 81.1% of the 936 correctly recognised focus words (51.9% of all focus words) the local word activation allowed us to identify the word before its last phone was available, and 64.1% of those words were already recognised one phone after the uniqueness point.

We investigate two types of predictors for deciding whether a word is considered as recognised before the end of its acoustic realisation. The first type is related to the absolute and relative values of the word activation,  $Act_{min}$  and  $\theta$ , respectively. The results show that the actual values of  $Act_{min}$  and  $\theta$  should not be set too high or too low, since both function as filters: The higher the values for both predictors, the fewer words are recognised, and vice versa. The second type of predictor is related to the number of phonemes of the word that have already been processed and the number of phonemes that remain until the end of the word. The results show that SpeM’s performance increases if the amount of evidence in support of a word increases and the risk of future mismatches decreases.

#### Chapter 5: A two-pass approach for handling OOVs in a large vocabulary recognition task

In Chapter 5, SpeM’s capability of recognising word-initial cohorts is used to address the problem of recognising a large vocabulary of over 50,000 city names within a telephone access spoken dialogue system. The experiments are conducted on spontaneous utterances within a joint domain of two spoken dialogue systems, a weather domain (Jupiter) and a flight reservation (Mercury) domain. Very large lexicons do not necessarily pose a problem for ASR systems, but the combination with a weak language model, which only has

virtually equal prior probabilities associated with each word, usually results in poor performance.

We adopt a two-stage framework in which only the 500 major cities are explicitly represented in the lexicons of both stages. In the first stage, we rely on an *unknown word* model encoded as a phone loop to detect out-of-vocabulary (OOV) city names (also referred to as rare city names). Of each rare city name the underlying phone graph is extracted. Subsequently, SpeM is used to extract words and word-initial cohorts from these phone graphs on the basis of a large fallback lexicon, to provide an  $N$ -best list of promising city name hypotheses. This  $N$ -best list is then inserted into the second stage lexicon for a subsequent recognition pass.

Experiments are conducted on a set of spontaneous telephone-quality utterances from both domains. These utterances were selected because they each contain a rare city name. The first experiment shows that SpeM is able to include nearly 75% of the correct rare city names in an  $N$ -best hypothesis list of 3000 city names.

In addition to the  $N$ -best lists of most likely words, the lexicon of the second stage also contains the so-called ‘base’ lexicon (which covers the other words in the utterance). In the second recognition experiment, we test two methods to create this base lexicon. The first method uses the same base lexicon as in the first stage, whereas the second method utilises a greatly pruned lexicon, based on the contents of the outputs of the first stage. The accuracy of the baseline recognition system (which excluded the  $N$ -best lists provided by SpeM) is 69.3%. Adding the  $N$ -best lists created by SpeM (method 1) increases the accuracy to 77.3%, a relative improvement of 11.5%. While the system with the pruned general lexicon (method 2) does not outperform the other system in terms of overall recognition error rate, it is able to correctly recognise up to 5% more rare city names. The final recognition results show that about one third of the rare city names that were found by SpeM are correctly recognised. So, work still remains to be done to improve on the second stage recogniser.

## Chapter 6: General discussion and concluding remarks

The central goal of this research was to narrow the gap that has existed between the research fields of human and automatic speech recognition. The research described in this thesis showed that despite the differences in goals and research methods, close parallels between ASR and HSR exist.

These close parallels between the two research fields were made explicit by the development of the end-to-end computational model of HSR created using techniques from the field of ASR, SpeM. The presented simulations and experiments carried out with SpeM showed that SpeM is able to simulate aspects of the human word recognition process while using the acoustic signal as input. The close parallels between the two research fields were further revealed by the development of the word activation scores used by SpeM. I showed

that it is possible to calculate a word-based continuous activation score from path-based ASR scores that is comparable to the activation scores used in HSR. The results showed that a left-to-right path-based decoding strategy as used in ASR systems (and in SpeM) is able to model the word-based competition effects found in behavioural studies of human speech recognition.

In conclusion, the most obvious contribution that ASR can make to HSR is to assist in the development of models that are able to account for the complete human speech recognition process from the acoustic analysis to the recognition of words in continuous speech. SpeM has proven to be a successful first step in that direction. The contribution of HSR to ASR is not yet as clear-cut. However, if a model of the complete human speech recognition process would exist, it could lead to interesting new ideas for the development of better ASR systems.





## Samenvatting (Summary in Dutch)

In het dagelijkse leven is spraak overal om ons heen, op de radio, televisie, en in menselijk contact. Mensen worden voortdurend geconfronteerd met nieuwe uitingen die sprekers selecteren uit een oneindige verzameling van mogelijke uitingen in een taal. En normaal gesproken hebben wij als luisteraars geen enkel probleem met het verstaan en begrijpen van de deze uitingen. Er zijn verschillende wetenschapsgebieden die onderzoek doen naar (delen van) het spraakherkenningsproces. In dit proefschrift concentreer ik me op twee van deze gebieden, namelijk de *automatische* spraakherkenning (ASH) en de *menselijke* spraakherkenning (MSH).

Ondanks dat beide onderzoeksgebieden sterk gerelateerd zijn (ze bestuderen immers allebei het spraakherkenningsproces), zijn hun doelen verschillend. In menselijke spraakherkenning is het doel om tot een volledig begrip te komen van het menselijke spraakherkenningsproces. Voor ASH is het doel het minimaliseren van het aantal fout herkende woorden. Als gevolg van de verschillen in doelen tussen de twee onderzoeksgebieden maken ze ook gebruik van verschillende onderzoeksmethoden.

Het belangrijkste verschil tussen beide onderzoeksgebieden voor dit proefschrift is echter dat zowel ASH als MSH claimt het volledige spraakherkenningsproces vanaf het akoestische signaal tot aan de herkende woorden te onderzoeken, terwijl de ASH dit als enige ook daadwerkelijk doet. Er bestaan geen computationele modellen van MSH die in staat zijn om het hele traject van het akoestische signaal naar de herkende woorden te simuleren. Het gedeelte van het spraakherkenningsproces dat in zo goed als alle computationele modellen ontbreekt, is de omzetting van het akoestische signaal naar een segmentele representatie. Met andere woorden, bestaande modellen van MSH kunnen geen echte spraak herkennen. Deze tekortkoming maakt het moeilijk om te testen of de assumpties die ten grondslag liggen aan de computationele modellen ook geldig zijn voor echte spraak.

Ondanks de grote verschillen tussen beide onderzoeksgebieden is er een groeiende interesse in mogelijke samenwerking. Het centrale doel van dit proefschrift is om de afstand tussen de wetenschapsgebieden van de ASH en de MSH te verkleinen en dus de twee onderzoeksgebieden dichtert tot elkaar te brengen. Deze uitdaging wordt gestart vanuit het oogpunt van de MSH door een *end-to-end* computationeel model van menselijke spraakherkenning te implementeren met behulp van technieken gebruikt in het veld van de ASH. Een end-to-end model van MSH is in staat om het hele spraakherkenningsproces vanaf het akoestische signaal tot aan de herkende woorden te simuleren.

### **Menselijke spraakherkenning**

In MSH worden laboratoriumexperimenten met menselijke luisteraars uitgevoerd om de eigenschappen van het menselijke spraakherkenningsproces bloot te leggen. Tijdens deze

experimenten worden verschillende soorten metingen verricht, zoals reactietijdmetingen en aantallen foutieve responsies. Op basis van deze metingen worden theorieën over bepaalde aspecten van het menselijke spraakherkenningsproces geformuleerd en verfijnd. Om deze theorieën vervolgens te testen worden er computationele modellen geïmplementeerd voor het simuleren en verklaren van menselijke spraakherkenning. Vrijwel alle bestaande computationele modellen van menselijke spraakherkenning modelleren slechts bepaalde aspecten van het menselijke spraakherkenningsproces. Voor dit proefschrift is het computationele model voor menselijke *woordherkenning Shortlist* van belang.

Volgens symbolische theorieën van MSH beelden menselijke luisteraars het binnenkomende akoestische signaal af op prelexicale representaties, bijvoorbeeld in de vorm van fonemen. Vervolgens worden deze prelexicale representaties afgebeeld op lexicale representaties, in de vorm van woorden. Volgens symbolische theorieën van MSH bestaat het menselijke spraakherkenningsproces dus uit twee niveaus: het prelexicale en het lexicale niveau. Een belangrijke eis voor een symbolisch computationeel model is dus een segmentele representatie van het spraaksignaal. Aangezien computationele modellen van MSH geen echte spraak kunnen verwerken, wordt er gebruik gemaakt van een handgemaakte segmentele representatie van het spraaksignaal als input van het lexicale niveau. Deze segmentele representatie van het spraaksignaal moet zo goed als foutloos zijn, in de zin dat deze perfect olijnt met de segmentele representaties van de woorden in het lexicon. Met andere woorden, in de meeste symbolische computationele modellen wordt het prelexicale proces dat de prelexicale representaties maakt op basis van het akoestische signaal verondersteld en is het niet fysiek aanwezig.

### **Automatische spraakherkenning**

De input van een automatisch spraakherkenningsstelsel bestaat uit het akoestische signaal. Gedurende het spraakherkenningsproces wordt het akoestische signaal eerst door een akoestische preprocessor bewerkt waar featurevectoren geëxtraheerd worden uit het akoestische signaal. Vervolgens worden deze featurevectoren vergeleken met de opeenvolging van akoestische modellen geassocieerd met de woorden in het interne lexicon van het ASH systeem. Voor iedere featurevector wordt berekend hoe goed het past op ieder van de akoestische modellen. Uiteindelijk wordt het woord dat correspondeert met de sequentie van akoestische modellen die het best past op de featurevectoren in de input als hypothese aangenomen. Een ASH systeem is alleen in staat die woorden te herkennen die voorkomen in zijn lexicon. Tenzij het ASH systeem een mogelijkheid heeft om een stukje input als 'geen woord' aan te merken, wordt een onbekend woord aan de input zal in principe altijd herkend worden als een van de woorden in het lexicon. Dit levert dus een foute herkenning op.

De meeste ASH systemen gebruiken een geïntegreerde zoekmethode voor het spraakherkenningsproces: alle beschikbare informatie (van de akoestische modellen, het lexicon en de taalmodellen) wordt tegelijkertijd gebruikt. Deze informatiebronnen vormen

samen de zoekruimte. Spraakherkenning is dus in feite het zoeken van het beste pad door de zoekruimte. Voor ieder pad (van een sequentie van woorden) door de zoekruimte wordt berekend hoe goed het past op de input; aan het einde van de input wordt het spoor terug gevolgd om de padhypothese met de beste score te bepalen en de woorden die op het pad liggen te identificeren. ASH systemen worden geëvalueerd in termen van accuraatheid, het percentage van de input uitingen dat goed herkend is, of in termen van word error rate (WER), het aantal geïnserteerde, gedeleerde en gesubstitueerde woorden gedeeld door het totale aantal woorden in de input.

ASH systemen zijn niet gebaseerd op theorieën van menselijke spraakherkenning zoals computationele MSH modellen. Daarom zal een ASH systeem nooit gebruikt kunnen worden als een end-to-end computationeel model van menselijke spraakherkenning. Een ASH systeem is echter wel in staat om een segmentele representatie van het spraaksignaal te maken. In het onderzoek beschreven in dit proefschrift wordt een aangepast ASH systeem, een automatische foonherkenner (AFH), gebruikt als het ontbrekende deel van huidige computationele modellen van MSH dat het akoestische signaal omzet in een segmentele representatie. Een AFH is vergelijkbaar met een standaard ASH systeem behalve dat het lexicon fonemen bevat in plaats van woorden. De output van een AFH is dan ook een representatie in de vorm van fonemen in plaats van woorden.

### **De opbouw van het proefschrift**

Naast een inleidend hoofdstuk dat hierboven in het kort samengevat is, staat bestaat dit proefschrift uit een viertal publicaties (hoofdstukken 2 tot en met 5) en een afsluitend hoofdstuk waarin de gevonden resultaten uit de vier publicaties aan elkaar gerelateerd worden en waarin de algemene conclusies getrokken op basis van dit proefschrift verwoord staan.

Het onderzoek beschreven in hoofdstuk 2 en 3 maakt duidelijk hoe MSH bij het bouwen van een end-to-end computationeel model van menselijke spraakherkenning kan profiteren van de technieken gebruikt in het veld van de ASH. Hoofdstuk 4 en 5 laten zien hoe een zoekmethode die gebaseerd is op menselijke spraakherkenning van nut kan zijn voor ASH. Hieronder volgen van ieder hoofdstuk kort de probleemstelling, doelstellingen, belangrijkste resultaten en conclusies.

### **Hoofdstuk 2: Het uitbreiden van Shortlist naar een end-to-end model van menselijke spraakherkenning**

Hoofdstuk 2 beschrijft de eerste poging om een end-to-end computationeel model te bouwen van MSH met behulp van technieken gebruikt in het veld van de ASH. Daartoe wordt Shortlist uitgebreid met een AFH die een segmentele representatie in de vorm van een foneemstring van het spraaksignaal maakt die vervolgens aan Shortlist als input wordt gegeven. Om te testen hoe goed dit gecombineerde model van een AFH en Shortlist echte spraak kan herkennen wordt een experiment uitgevoerd. Aan het gecombineerde model

worden 10.510 uitingen bestaande uit een Nederlandse plaatsnaam of ‘ik weet het niet’ uitgesproken in isolatie aangeboden.

De experimenten laten duidelijk de consequenties zien van sommige vereenvoudigde aannames die gemaakt worden in Shortlist en andere modellen van MSH. Verder wordt duidelijk in hoeverre deze aannames aangepast moeten worden om tot een end-to-end model van menselijke spraakherkenning te komen dat in staat is om spraak te herkennen vanaf het akoestische signaal. De grootste tekortkoming van het gecombineerde model van een AFH en Shortlist is dat het ‘harde’ beslissingen neemt op het niveau van de inputfonemen. Shortlist verlangt een een-dimensionale foonstring als input. Dit betekent dat de AFH gedwongen wordt om ‘harde’ beslissingen te nemen met betrekking tot welke fonemen er in het signaal aanwezig zijn, gebaseerd op alleen maar de akoestiek. De tweede tekortkoming is de eenvoudige zoekmethode gebruikt in Shortlist: een deletie of een insertie van een foneem in de input veroorzaakt een foutieve ophijning van alle volgende fonemen met de woorden in het lexicon.

Concluderend, een combinatie van een AFH en Shortlist leidt niet tot een end-to-end model van MSH dat goed genoeg om kan gaan met echte spraakinput, ondanks dat Shortlist een succesvol computationeel model is van een specifiek aspect van het menselijke spraakherkenningsproces.

### Hoofdstuk 3: Hoe zou een spraakherkenner moeten werken?

Zoals eerder al is opgemerkt is er weinig communicatie tussen de onderzoeksgebieden van de ASH en de MSH. In hoofdstuk 3 wordt gesuggereerd dat dit gebrek aan communicatie komt doordat beide onderzoeksgebieden focussen op *hoe* spraak herkend kan worden (dit worden het algoritmische en het implementatieniveau genoemd door Marr (1982)) in plaats van *welke functies* er nodig zijn om spraak te kunnen herkennen en *waarom* deze functies nodig zijn (het computationele niveau volgens Marr). In hoofdstuk 3 wordt een computationele analyse gegeven van het spraakherkenningsproces die de parallellen tussen beide vakgebieden duidelijk laat zien. Deze parallellen worden verder verduidelijkt door de implementatie van een nieuw computationeel model van MSH, SpeM (Speech-based model of HSR), dat gebouwd is met technieken van ASH. SpeM is gebaseerd op de theorie die de basis is voor aan Shortlist. In tegenstelling tot huidige computationele modellen van MSH is SpeM met redelijk veel succes in staat om woorden te herkennen op basis van het akoestische signaal.

Het probleem van de ‘harde’ beslissingen op het niveau van de inputfonemen waar het gecombineerde model in hoofdstuk 2 problemen mee had is in SpeM opgelost door de een-dimensionale foonstring te vervangen door een probabilistische foongraaf die meerdere foonstring hypothesen naast elkaar bevat. Door deze representatie van het spraaksignaal wordt op een natuurlijke manier de beslissing over een foneem verschoven naar een later punt in het woordherkenningsproces. De tweede tekortkoming van het gecombineerde

model beschreven in hoofdstuk 2, de lexicale zoekmethode, is opgelost in SpeM door gebruik te maken van een zoekmethode gebaseerd op dynamische programmeermethoden. Hierdoor worden inserties en deleties in de input ten opzichte van de woorden in het lexicon beter afgehandeld.

Er worden twee typen experimenten uitgevoerd met SpeM. Het eerste experiment is een kopie van het experiment uitgevoerd met het gecombineerde systeem van hoofdstuk 2. Dit experiment laat zien dat SpeM veel beter in staat is om de juiste woorden te herkennen op basis van de akoestiek dan het gecombineerde systeem. Dit komt voornamelijk doordat SpeM geen een-dimensionale foonstring gebruikt maar een probabilistische foongraaf.

In het tweede type experiment wordt de kracht van SpeM met betrekking tot het simuleren van menselijke spraakherkenning onderzocht. Voor de simulaties van menselijke spraakherkenning zijn *woordactivaties* nodig, terwijl SpeM *padgebaseerde scores* berekent. In hoofdstuk 3 wordt een methode gepresenteerd om de padgebaseerde scores van SpeM om te schrijven naar woordgebaseerde activaties. De drie simulaties uitgevoerd met SpeM lieten zien dat SpeM, met het akoestische signaal als input, in staat is om dezelfde aspecten van het menselijke spraakherkenningssysteem te simuleren als Shortlist, dat een foneemstring als input heeft.

#### Hoofdstuk 4: ‘Vroege herkenning’ van polysyllabische woorden in continue spraak

Mensen zijn in staat om een woord te herkennen voordat de akoestische realisatie van het woord compleet is. Dit in tegenstelling tot ASH systemen die een woord pas na een frase of een zin herkennen om de herkenprestatie van het ASH systeem te optimaliseren. Verder, in tegenstelling tot de traditionele geïntegreerde zoekmethoden gebruikt in ASH systemen maakt SpeM gebruik van een incrementele zoekmethode. Deze incrementele zoekmethode geeft een geordende lijst van hypothesen voor ieder moment in de tijd gedurende het spraakherkenningsproces aan de output en is daarom in staat om een woord *voor* het einde van zijn akoestiek te herkennen. In hoofdstuk 4 wordt onderzocht hoe de incrementele zoekmethode van SpeM gebruikt kan worden voor het herkennen van een woord voordat de akoestiek behorende bij het woord gehoord is. Dit noemen we ‘vroege herkenning’.

Experimenten op 1.463 polysyllabische focuswoorden in 885 uitingen laten zien dat 64.0% van de focuswoorden correct herkend waren aan het einde van uiting. 81.1% van deze correct herkende focuswoorden (51.9% van alle focuswoorden) had een woordactivatiescore die hoger was en bleef dan die van de andere hypothesen *voor* het einde van het akoestische signaal dat hoort bij het woord. 64.1% van de correct herkende focuswoorden was 1 foneem na het uniekheidspunt al herkend.

Verder zijn er twee types voorspellers onderzocht die gebruikt kunnen worden *tijdens* het herkenproces om te bepalen hoe groot de kans is dat de hypothese correct is. De eerste is gebaseerd op de absolute en de relatieve woordactivatie berekend door SpeM,

respectievelijk de minimum activatie ( $Act_{min}$ ) en  $\theta$  genaamd. De resultaten laten zien dat de waarden van  $Act_{min}$  en  $\theta$  niet te hoog noch te laag moeten zijn aangezien ze beide werken als filters: hoe hoger de waarden van de twee voorspellers, hoe minder woorden er herkend worden en vice versa. Het tweede type voorspeller is gerelateerd aan het aantal fonemen van het woord dat al verwerkt is en hoeveel fonemen er nog niet verwerkt zijn tot aan het einde van het woord. Deze resultaten laten zien dat de prestatie van SpeM verbetert als de hoeveelheid fonemen die al verwerkt zijn groter wordt en het aantal fonemen tot aan het einde van het woord dat nog niet verwerkt is kleiner wordt.

## Hoofdstuk 5: Een twee-staps methode voor het verwerken van OOVs in een herkenningstaak met een groot lexicon

In hoofdstuk 5 wordt het vermogen van SpeM om woorden voor het einde van de bijbehorende akoestiek te herkennen aangewend om woordinitiële cohorten te herkennen die vervolgens gebruikt worden om het probleem van woorden die niet in het lexicon voorkomen, en dus niet herkend kunnen worden, aan te pakken. Meer specifiek, hoofdstuk 5 behandelt het probleem van het herkennen van de woorden in een groot lexicon met meer dan 50.000 plaatsnamen in twee telefoongestuurde dialoogsystemen: een weerinformatiesysteem (Jupiter) en een vluchtreserveringssysteem (Mercury).

Het probleem met extreem grote lexicons (in combinatie met zwakke taalmodellen) in spraakherkenningssystemen is dat ze zorgen voor veel foute herkenningen. De oplossing voorgesteld in hoofdstuk 5 is om een twee-staps methode te gebruiken die in beide stappen een spraakherkenningssysteem gebruikt met een klein lexicon. Daartoe worden in het spraakherkenningssysteem van de eerste stap alleen de 500 hoogst-frequente plaatsnamen expliciet opgenomen in het lexicon. Met behulp van een *onbekend-woordmodel* worden in de spraakherkenner van de eerste stap alle woorden die niet in het lexicon voorkomen, dus de laag frequente woorden, of ‘infrequente plaatsnamen, afgevangen en gemerkt. Van deze infrequente plaatsnamen wordt de onderliggende foonstructuur in de vorm van een foongraaf opgeslagen. Vervolgens wordt met behulp van SpeM op basis van het grote lexicon met 50.000 woorden een selectie gemaakt van de meest waarschijnlijk plaatsnaamhypotheses op basis van de foongraaf onderliggend aan de infrequente plaatsnaam. SpeM levert deze meest waarschijnlijke plaatsnaamhypotheses in de vorm van een *N*-best lijst. Tot slot wordt deze *N*-best lijst toegevoegd aan het lexicon van de spraakherkenner in de tweede stap.

Experimenten worden uitgevoerd op een set van spontane uitingen van telefoonkwaliteit uit zowel het Mercury als het Jupiter domein. Iedere uiting in de testset bevat één zeldzame plaatsnaam. Het eerste experiment laat zien dat SpeM in staat is om bijna 75% van de goede zeldzame plaatsnamen in de 3000-best lijst te selecteren. Naast de *N*-best lijst gegenereerd door SpeM bevat het lexicon van de spraakherkenner van de tweede stap ook een zogenaamd ‘basis’lexicon (dat alle overige woorden van de uiting bevat). In het tweede

herkenexperiment worden twee methodes om het basislexicon te maken getest. De eerste methode gebruikt hetzelfde lexicon als de spraakherkenner in de eerste stap; de tweede methode gebruikt een sterk gereduceerd lexicon gebaseerd op de output van de eerste herkenstap.

Het baseline herkensysteem (zonder de *N*-best lijsten gegenereerd door SpeM) is in staat om 69.3% van alle woorden correct te herkennen. Het toevoegen van de *N*-best lijsten van SpeM (methode 1) verhoogt het percentage goed herkende woorden naar 77.3%, een relatieve verbetering van 11.5%. Ondanks dat het systeem met het gereduceerde lexicon (methode 2) een gelijke herkenprestatie haalt, is het in staat om 5% meer van de zeldzame plaatsnamen te herkennen. Het laatste experiment laat zien dat ongeveer eenderde van de zeldzame plaatsnamen die gevonden waren door SpeM correct herkend worden door gebruikmaking van de twee-staps methode. Het is duidelijk dat de herkenprestatie van de spraakherkenner van de tweede stap verbeterd moet worden.

## Hoofdstuk 6: Discussie en algemene conclusies

Het centrale doel van dit proefschrift was om de grote verschillen tussen de onderzoeksgebieden van de automatische en menselijke spraakherkenning te verkleinen. Het in dit proefschrift beschreven onderzoek heeft laten zien dat ondanks de grote afstand tussen beide onderzoeksgebieden in doelen en onderzoeksmethoden er veel parallellen zijn tussen ASH en MSH.

Deze parallellen tussen beide onderzoeksgebieden zijn ten eerste expliciet gemaakt door de ontwikkeling van het end-to-end computationele model van MSH, SpeM, dat geïmplementeerd is met gebruikmaking van technieken van ASH. De simulaties en experimenten uitgevoerd met SpeM hebben laten zien dat SpeM in staat is om het menselijke spraakherkenningsproces te simuleren vanaf het akoestische signaal tot aan de uiteindelijk herkende woorden. Ten tweede zijn de duidelijke parallellen zichtbaar gemaakt door de ontwikkeling van de woordactivatiescores die gebruikt worden door SpeM. In dit proefschrift hebben we laten zien dat het mogelijk is om een woordgebaseerde maat te berekenen op basis van de padgebaseerde scores die gebruikt worden in standaard ASH systemen. De resultaten van de experimenten uitgevoerd met SpeM laten zien dat SpeM in staat is om de woordgebaseerde competitie-effecten die gevonden worden in gedragsstudies naar menselijke spraakherkenning te simuleren.

Concluderend, de meest duidelijke bijdrage die ASH kan leveren voor MSH is assistentie bij het ontwikkelen van computationele modellen die het hele spraakherkenningsproces vanaf het akoestische signaal tot aan de uiteindelijk herkende woorden kunnen simuleren. SpeM is een succesvolle stap in de goede richting. De bijdrage van MSH voor ASH is niet zo duidelijk. Echter, als er een computationeel model van MSH zou bestaan, zou dit kunnen leiden tot interessante nieuwe ideeën voor de ontwikkeling van betere ASH systemen.





## Curriculum vitae

Odette Scharenborg was born on the 26<sup>th</sup> of April, 1977 in Groenlo, the Netherlands. For her primary education she attended the St. Willibrordus Primary School in Groenlo. She subsequently attended secondary school at the R.K. S.G. Marianum in Groenlo, from which she graduated in 1995. In that same year, she started to study Language and Speech Technology at the Radboud University Nijmegen, the Netherlands. She received her Master's diploma in 2000 having specialised in automatic speech recognition. From April to December 2000, she worked as a junior researcher within the framework of the European SMADA (Speech Driven Multi-model Automatic Directory Assistance) project at the Department of Language and Speech at the Radboud University Nijmegen. From January 2001 to June 2005 Odette Scharenborg was employed as a PhD student at the Department of Language and Speech at the Radboud University Nijmegen. During this period she spent two months in 2002 at the Medical Research Council – Cognition and Brain Sciences Unit (MRC-CBU), Cambridge, UK, as a visiting researcher. In November 2003, she was co-organiser of the workshop “Innovative approaches bridging automatic and human speech recognition”. In 2004, she spent three months at the Spoken Language Systems Group, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, as a visiting researcher. This thesis is a result of the work carried out at the Department of Language and Speech in close collaboration with the Max Planck Institute of Psycholinguistics, Nijmegen, at the MRC-CBU, and at MIT. Odette Scharenborg is currently employed as a researcher at the Department of Language and Speech, Radboud University Nijmegen.



## List of publications

### **This thesis consists of the following publications:**

- Scharenborg, O., ten Bosch, L., Boves, L., Norris, D. (2003). Bridging automatic speech recognition and psycholinguistics: Extending Shortlist to an end-to-end model of human speech recognition (L). *Journal of the Acoustical Society of America*, 114 (6), 3032-3035.
- Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M. How should a speech recognizer work? *Accepted for publication in Cognitive Science*.
- Scharenborg, O., ten Bosch, L., Boves, L. 'Early recognition' of polysyllabic words in continuous speech. *Resubmitted to Computer Speech and Language*.
- Scharenborg, O., Seneff, S., Boves, L. A two-pass approach for handling OOVs in a large vocabulary recognition task. *Submitted to Computer Speech and Language*.

### **Other publications not included in this thesis:**

- Scharenborg, O. (2005). Parallels between HSR and ASR: How ASR can contribute to HSR. *To appear in the proceedings of Interspeech*, Lisbon, Portugal.
- Scharenborg, O., Seneff, S. (2005). A two-pass strategy for handling OOVs in a large vocabulary recognition task. *To appear in the proceedings of Interspeech*, Lisbon, Portugal.
- Ten Bosch, L., Scharenborg, O. (2005). ASR decoding in a computational model of human word recognition. *To appear in the proceedings of Interspeech*, Lisbon, Portugal.
- Scharenborg, O., Boves, L., ten Bosch, L. (2004). 'On-line early recognition' of polysyllabic words in continuous speech. *Proceedings of the Tenth Australian International Conference on Speech Science & Technology*, Sydney, Australia, cdrom.
- Scharenborg, O., ten Bosch, L., Boves, L. (2003). 'Early recognition' of words in continuous speech. *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, St. Thomas, US Virgin Islands, cdrom.
- Scharenborg, O., McQueen, J.M., ten Bosch, L., Norris, D. (2003). Modelling human speech recognition using automatic speech recognition paradigms in SpeM. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 2097-2100.
- Scharenborg, O., ten Bosch, L., Boves, L. (2003). Recognising 'real-life' speech with SpeM: A speech-based computational model of human speech recognition. *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 2285-2288.

- Scharenborg, O., Boves, L. (2002). Pronunciation variation modelling in a model of human word recognition. *Proceedings of the ISCA Workshop on Pronunciation Modeling and Lexicon Adaptation*, Estes Park, CO, pp. 65-70.
- Scharenborg, O., Boves, L., de Veth, J. (2002). ASR in a human word recognition model: Generating phonemic input for Shortlist. *Proceedings of ICSLP*, Denver, CO, pp. 633- 636.
- Scharenborg, O., Sturm, J., Boves, L. (2001). Business listings in automatic directory assistance. *Proceedings of Eurospeech*, Aalborg, Denmark, pp. 2381-2384.
- Scharenborg, O., Bouwman, G., Boves, L. (2000). Connected digit recognition with class specific word models. *Proceedings of the COST249 Workshop on Voice Operated Telecom Services*, Ghent, Belgium, pp. 71-74.