Cognitive Science 29 (2005) 867–918

How Should a Speech Recognizer Work?

Odette Scharenborg[a], Dennis Norris[b], Louis ten Bosch[a], James M. McQueen[c]

[a]*Radboud University Nijmegen, The Netherlands*
[b]*Medical Research Council Cognition and Brain Sciences Unit, Cambridge, UK*
[c]*Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands*

Received 4 May 2004; received in revised form 21 December 2004; accepted 13 May 2005

Abstract

Although researchers studying human speech recognition (HSR) and automatic speech recognition (ASR) share a common interest in how information processing systems (human or machine) recognize 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 focused on the mechanics of *how* speech can be recognized. In Marr's (1982) terms, emphasis has been on the algorithmic and implementational levels rather than on the computational level. In this article, we provide a computational-level analysis of the task of speech recognition, which reveals the close parallels between research concerned with HSR and ASR. We illustrate this relation by presenting a new computational model of human spoken-word recognition, built using techniques from the field of ASR that, in contrast to current existing models of HSR, recognizes words from real speech input.

Keywords: Human speech recognition; Automatic speech recognition; Spoken-word recognition; Computational modeling

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 recognized. It might seem, therefore, that this common goal would foster close links between the disciplines. However, although 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, recognize 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

Requests for reprints should be sent to Odette Scharenborg, P.O. Box 9103, 6500 HD Nijmegen, The Netherlands. E-mail: o.scharenborg@let.ru.nl

in both disciplines. In his book *Vision* (1982), Marr argued 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, whereas the implementational level is concerned with how representations and algorithms can be realized physically. From an information processing perspective, Marr suggested 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 suggested 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 recognizing human speech. In presenting this analysis, we relate the computational level to the different algorithms used in ASR and HSR. Although ASR uses vocabulary such as dynamic programming (DP) and preprocessing, and HSR is described in terms of lexical competition and auditory perception, we show that most of these terms have direct homologs 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 ASR. This new model, called *Speech-based Model* (SpeM), can be considered to be an implementation of the shortlist model (Norris, 1994) with one important difference from shortlist: SpeM can recognize real speech.

1.1. A common goal

In HSR research, the goal is to understand how we, as listeners, recognize 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 this 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 to computer speech systems and human listeners equally well. Whether the speech

recognizer 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 to recognize spoken words?

1.2. Human speech recognition

Explanatory theories in HSR have generally focused 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 are no constraining data, or no a priori reason to prefer one implementation to another. That is, these are primarily piecemeal 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 recognizer? More important, 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.

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 recognizer 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 the following for details), unencumbered by any considerations of psychological plausibility. An ASR system must be able to recognize speech tolerably well under favorable conditions, but nothing in the behavior of such a system needs to map onto any observable human behavior, 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.

1.4. Summary

In a sense, therefore, we have two complementary models of speech recognition. HSR models explain at least some human behavior, 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 recognizing spoken words with the acoustic signal as starting point. In contrast, ASR models can recognize speech, but offer little in the way of explaining human behavior. HSR and ASR are, however, not complementary in the set-theoretic sense of being nonoverlapping. For either type of model to be a success, it has to be consistent with a computational-level description of the problem of recognizing 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.

2. Computational analysis of word recognition

2.1. Prelexical and lexical levels of processing

Two acoustic realizations 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 recognizer 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 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 organization 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 sublexical representations. Learning to recognize a word reliably will require exposure to a large number of acoustic realizations of that particular word. That is, a model that simply stores multiple episodes of words has to learn each word independ-

ently. Nothing the model learns about recognizing one word will make it any better at recognizing previously unencountered words.[1]

A similar issue of generalization occurs across speakers. It is rather unclear how an episodic word recognition system could robustly recognize 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 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 recognize words spoken by the new speaker. To take account of the systematicity in pronunciation, an episodic model would first of all have to be able to analyze both the input and the episodic lexical representations in terms of their sublexical components and then have 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, McQueen, and Cutler (2003) showed that listeners can quickly learn that a speaker produces a particular phoneme in an unusual manner; moreover, McQueen, Cutler, and Norris (2005) have shown that this knowledge generalizes 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 generalization 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 generalization 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 generalize to new words and speakers. A drawback to this type of theory, 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 modeling the acoustic–phonetic information in the speech signal in terms of a limited number of subword units and, thus, offer the possibility of 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 generalize 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, 2005, 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 (Gaskell & Marslen-Wilson, 1997); and context-sensitive allophones in PARSYN (Luce, Goldinger, Auer & Vitevitch, 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 subword statistical models that encode the expected variation in the signal. These subword models could in principle represent several types of speech segments (e.g., phones[2], 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 subword units. Most mainstream mid- and large-vocabulary ASR systems are based on statistical phone models (see, e.g., Juang & Furui, 2000; Lesser, Fennell, Erman, & Reddy, 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, but eliminate 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 modeling 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 and Juang (1993).

Fig. 1 shows a graphical representation of an HMM consisting of three states (indicated by the circles in Fig. 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 (indicated as *b* in Fig. 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 neighboring recognition units or when people with different regional accents produce that specific recognition unit. The probability density functions are estimated

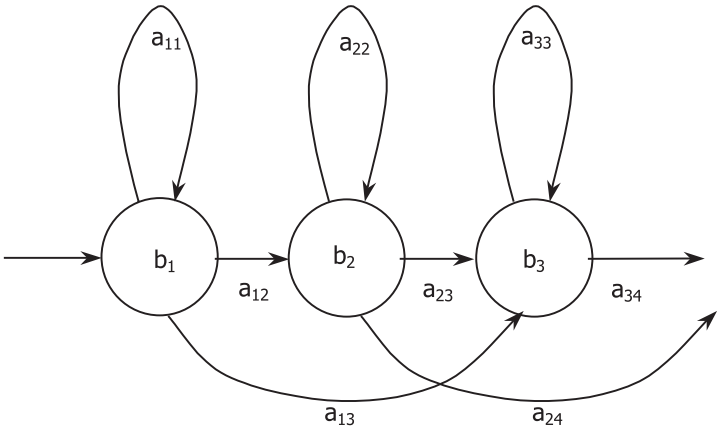

Fig. 1. A graphical representation of a Hidden Markov Model consisting of three states.

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 modeled by a set of transition probabilities (indicated by a in Fig. 1), namely,

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

A very common procedure for training acoustic models maximizes the likelihood that a given acoustic signal has been generated by a given acoustic model; more precisely, it maximizes $P(X|S)$ (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 subword models have the same advantage as HSR models with prelexical representations: New words can be learned simply by learning the appropriate sequence of subword models, and such knowledge will automatically generalize to new tokens of the word. In fact, once a sufficient range of subword models has been trained, new words can be recognized 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 ViaVoice and ScanSoft's Dragon Dictate work. The recognizer only requires exposure to a representative sample of speech to achieve accurate recognition of a large vocabulary.

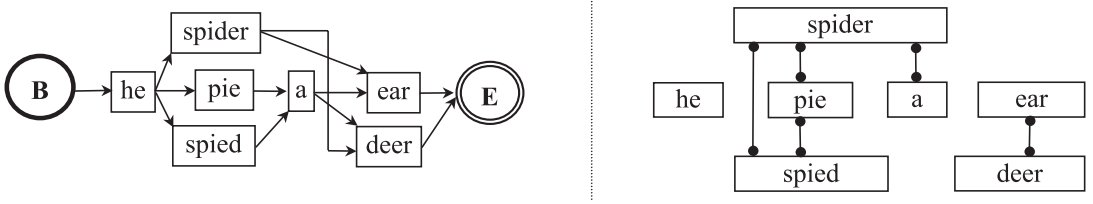

Fig. 2. The left panel shows an example of a word lattice as used in automatic speech recognition; the right panel shows the competition process that occurs in human speech recognition.

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 previously that for successful speech recognition, a prelexical level is needed 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 Fig. 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 Fig. 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 subword unit), the structure of the word in the lexicon in terms of the subword 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 recognizer. 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 Fig. 2) and one end node (denoted E in Fig. 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.

2.2. *Cascaded prelexical level*

An ideal speech recognizer would be able to recognize 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 recognized: On the basis of results from a number of different listening tasks, Marslen-Wilson (1987) estimated this lag to be only 200 msec (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, Dahan, and Cutler (2003) 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, Blumstein, & Burton, 1994; Davis, Marslen-Wilson, & Gaskell, 2002; Gow, 2002; Marslen-Wilson & Warren, 1994; McQueen, Norris, & Cutler, 1999; Salverda, Dahan, & McQueen, 2003; Spinelli, McQueen, & Cutler, 2003; Tabossi, Collina, Mazzetti, & Zoppello, 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, Magnuson, & Tanenhaus, 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 offline (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 recognize dictated speech (e.g., ViaVoice and Dragon Dictate) can often produce results shortly after words have been uttered. However, the solution that is hypothesized 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 hypothesized 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 recognize a word until several following words have been heard (Bard, Shillcock, & Altmann, 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 previously, 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 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, Laureys, Van Compernolle, & Van Hamme, 2003). The first step in-

volves 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 recognize speech rapidly, both HSR models and ASR systems have converged on cascaded processing algorithms.

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 2.1).

The matching process necessarily depends on the form of the prelexical representations. In an HSR model such as 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 recognized. 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 recognized 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 Distributed Cohort Model, Gaskell & Marslen-Wilson, 1997; and PARSYN, Luce et al., 2000) there is therefore some kind of competition mechanism that 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, Burani, & Scott, 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, Norris, & Cutler, 1994; Norris, McQueen, & Cutler, 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 (2005).

The competition mechanism in HSR models helps them solve what we 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 because 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, Cutler, Briscoe, & Norris, 1995). This means that when the recognizer 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 maximizes 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 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 textbook 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, 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's rule and is related to (a) the probability that the acoustic signal is produced given the word ($P[X|W]$), and (b) the prior probability of the word—usually based on its frequency of occurrence; $P(W)$.

Thus, although 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.

2.4. Continuous speech recognition

So far, our computational analysis of spoken-word recognition has focused on the task of identifying isolated words. However, the real task facing a listener, or an automatic speech recognizer, 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, *speechiscomparabletohandwrittentextswithoutspaces* (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* ([ʃɪpɪŋkwɑɪəri]). 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), whereas the evaluation process in HSR is at the word level (e.g., competition between individual lexical hypotheses). This difference is illustrated in Fig. 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, whereas 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 one another. Also, because words with more bottom-up support have more activation, the competition process will tend to favor 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): (a) the probability that the acoustic signal is produced, given the word sequence ($P(X|Path)$, in which *Path* denotes a word or word sequence); (b) the prior probability of each word (based on its frequency of occurrence); and (c) 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 DP algorithm instead of an interactive activation model, and Fig. 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.

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) argued 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 boundaries (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 penalized (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, Demuth, & McQueen, 2002).

Norris et al. (1997) suggested that an important benefit of the PWC was that it would help solve the problem of recognizing 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 recognizers 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. (1997) argued, a competition-based model such as shortlist, if *fourf* is not stored as a word form in the lexicon, will tend to recognize such a sequence as *They metaphor f time.* This is clearly inadequate. What is required is a mechanism that 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 penalized 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., 1997, 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, Minematsu, Hashimoto, and Iwano (2001) 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 dramat-

ically, 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 labeled as *out-of-vocabulary* and usually not processed further (there are exceptions, for instance, the second pass in HDLA does process the entries labeled 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), whereas 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.

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, McQueen, and Cutler (2000) argued that there was no psycholinguistic data that required the postulation of a lexical feedback mechanism, and they also argued that some data (that of Pitt & McQueen, 1998) challenged HSR models such as TRACE, which have feedback. Furthermore, Norris et al. (2000) 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 sublexical 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, McMurray, Tanenhaus, & Aslin, 2003; McQueen, 2003; Norris et al., 2003; Samuel, 2001; Samuel & Pitt, 2003).

ASR systems do not use the kind of online 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, 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. Because 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) argued, feedback can be of benefit in retuning prelexical representations. The experiments that Norris et al. (2003) reported 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 categorized 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 online feedback (where the lexical level influences prelexical processing as speech is being input to the recognizer) and offline 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 toward 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 online feedback does not usually arise in ASR, because of the integration of the prelexical and lexical-level information into one decoding structure.

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 recognizers 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 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 (see also, Scharenborg, McQueen, ten Bosch, & Norris, 2003; Scharenborg, ten Bosch, & Boves, 2003b). SpeM is a new implementation of the shortlist model (Norris, 1994) developed using ASR techniques. In contrast to existing HSR models, SpeM can recognize words from real speech input.

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 3.5 shows, 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 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 recognize words given acoustic input (rather than a handcrafted symbolic description of the speech signal). We present simulations (Sections 4 and 5) showing that SpeM can indeed recognize 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 recognizer should work.

The architecture of the SpeM model is shown in Fig. 3. SpeM consists of three modules. The first module is an automatic phone recognizer (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 3.1) using statistical acoustic models (see Section 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) 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 recognized (see Section 3.5). Each of the key computational problems identified in Section 2 are dealt with in the SpeM model, as described in the following.

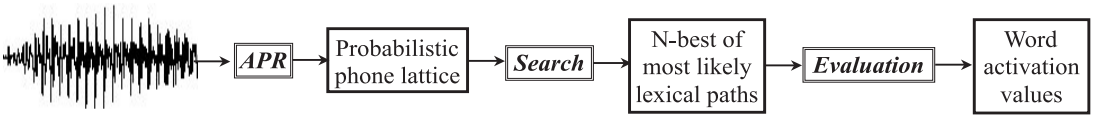

Fig. 3. Overview of the SpeM model.

3.1. Prelexical and lexical levels of processing

In Section 2.1, we argued that a speech recognizer 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). Fig. 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 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 [ɑs] (the Dutch word *as,* "ash") are shown.

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 Fig. 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

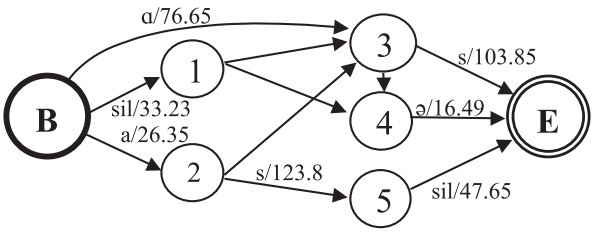

Fig. 4. A graphical representation of a weighted probabilistic input phone lattice. For sake of clarity, not all phones and acoustic costs are shown.

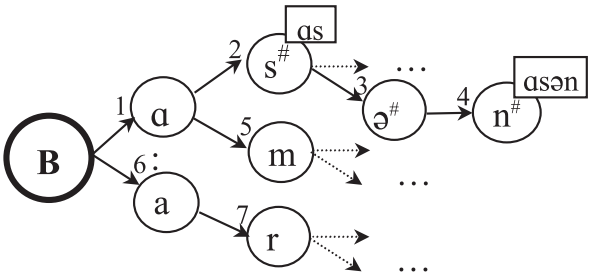

Fig. 5. A graphical representation of the beginning of a lexical tree.

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 [ɑs] is an example of an embedded word, because the node labeled with [ɑs] in the lexical tree (Fig. 5, Node 2) has outgoing arcs (thus in this case the phonemic transcription [ɑs] also represents a word-initial cohort). As described in more detail later, 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, although implemented in SpeM, are not used in the simulations described in this article.

3.2. Cascaded prelexical level

Rapid online 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 (although, as Norris, 2005, 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. 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 2.3), and the evaluation module. The search module computes the bottom-up goodness of fit of different lexical hypotheses to this input, whereas the evaluation module acts to compare those hypotheses with each other (see Section 3.5). During the search process, the best path (the opti-

mal sequence of words) is derived using a time-synchronous Viterbi search through a search space that 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 2.1 and 2.3). As noted earlier, Viterbi search is simply an efficient method for finding the best path through a lattice.

The words hypothesized 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 this input. Whenever the mismatch between the hypothesized 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 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 2.3). For instance, when Node 2 (Fig. 5) is accessed, not only is the word [ɑs] activated but also all words that have [ɑs] 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 3.1); it is the probability that the acoustic signal is produced given the phone ($P(X|Ph)$, in which *Ph* denotes a phone (see Section 2.3).
- *Symbolic phone matching cost:* This is the cost associated with this 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 3.4.
- *History cost:* This is the total cost of the path up to the mother node, that is, the search-space node from which this 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 that 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 the additional overhead of modeling 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 (contain-

ing, 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 these 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 preset 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 (Fig. 4) and the root node of the lexical tree (Fig. 5). As already indicated, the search algorithm is time synchronous. First Node 1 of the phone lattice is evaluated:

1. 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.
2. 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).
3. 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 Fig. 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's rule (see Section 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* ([ɑs]) 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 Nodes B and 3 in Fig. 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.

3.4. Segmentation of continuous speech

In Section 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 penalized. 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 hypothesized 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 hypothesized. 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 Fig. 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 nongarbage word has begun, there is a PWC check on the sequence of garbage symbols.

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 this input, they are not normalized relative to each other. That is, each path score is independent of all other path scores. As we discussed in Section 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 model 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 an *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. Because 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. This 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, McQueen, & Smits, 2004, 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 that 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's rule, to the probability *P*(*W*|*X*) of observing a word *W,* given the signal *X.* Bayes's 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's 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, ten Bosch, & Boves, 2003a).

In this 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. (2003a). The key components of these computations, however, are as follows:

The probability of word *W* given the acoustic signal *X* is based on Bayes's rule:

$$P(W \mid X) = \frac{P(X \mid W) \bullet P(W)}{P(X)} \tag{1}$$

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 normalization of the a posteriori probability *P*(*W*|*X*), *P*(*X*) is often evaluated as follows:

$$P(X) = \sum P(X \mid W) \bullet P(W) \tag{2}$$

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 output by SpeM. Instead, we evaluate *P*(*X*) as follows:

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

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 $P(X|W) \bullet P(W)$ is normalized on the basis of the number of phones along the path that is making up the word sequence *W.* This normalization is very similar to the normalization of acoustic scores applied in the evaluation of confidence measures, the difference being that these confidence normalizations are often based on the number of frames instead of on the number of phones. This option has been chosen because, 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 (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 hypothesized paths. Incorporating this into Equation 1 leads to

$$P(W(n) \mid X_w(t)) = \frac{P(X_w(t) \mid W(n)) \bullet P(W(n))}{P(X_w(t))} \tag{4}$$

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/, that is, 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 article, $P(W(n))$ is the same for all cohorts and all words—that is, all words and cohorts have equal 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 4

$$P(Path \mid X_p(t)) = \frac{P(X_p(t) \mid Path) \bullet P(Path)}{P(X_p(t))} \tag{5}$$

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 all simulations are based on a unigram language model.

Both Equations 4 and 5 deal with normalization over time. The probabilities they compute are not yet normalized 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. To make an across-path normalized 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 normalization is an activation measure that is both normalized over time and across paths.

Although an across-path normalization is necessary, it is not necessary to use the entire probability mass that is present in the full word lattice for the normalization. Instead, it is sufficient to normalize 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 that which the normalization is based on, 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-normalized word activation ($Act(W(n))$) for a word *W* is therefore calculated as follows:

$$Act(W(n)) = \frac{P(W(n) \mid X_w(t) \bullet P(Path \mid X_p(t))}{\Pr Mass} \tag{6}$$

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 normalized 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 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, Schlueter, Macherey, & Ney, 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, Boves, & Koolwaaij, 2000; Wessel et al., 2001). The confidence measure is the degree of certainty that an ASR system has that it has recognized 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 in 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's 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. used 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 showed 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's rule to provide an estimate of the probability that a listener would identify that word given that input—an estimate that changes over time as the speech input unfolds.

3.6. No feedback from the lexical level to the prelexical level

In Section 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 online 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.

3.7. 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 (because all matching candidate words in the lexical tree are considered in parallel during the search and then compared during evaluation) and the segmentation problem (because 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 penalizes paths that include impossible words (garbage sequences without vowels),

but does not penalize those with garbage sequences that 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 2. An obvious question now arises: How does SpeM perform?

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 recognize 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 multispeaker 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 because 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 idealized 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 2.1). In particular, if there is some degree of phonetic mismatch between this 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 interrelated 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 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 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 recognizer 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 Fig. 5). The lexical representations in shortlist are also sequences of phonemes. It might therefore appear that both of these models would be unable to recognize 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 lookup (as shortlist does). Instead, it uses a DP algorithm that is able to align two strings of different lengths (see Section 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). In the context of a DP algorithm, which tolerates misalignment between the input and canonical lexical representations, it is necessary to include a mechanism that 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 recognize words in spite of insertions and deletions. We compared a version of the model with the insertion/deletion penalties (see Section 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 lookup 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).

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 (Sturm, Kamperman, Boves & den Os, 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 3.1). Because shortlist takes a symbolic de-

scription of the speech signal as input, it is not able to recognize 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 Dutch Directory Assistance 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 optimized on a subset of 100 utterances from this test corpus. Parameter values in both shortlist and SpeM were adjusted to maximize the number of correctly recognized words in each case. In shortlist, the optimized parameter was the mismatch parameter (see Scharenborg, ten Bosch, Boves, & Norris, 2003; also for related shortlist simulations).

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 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

Table 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

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

Note. The three versions of SpeM that used a probabilistic prelexical representation include: 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).

performance is solely contributable to the implementation of the search because 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 implemention in SpeM is somewhat better able to deal with the variability in real speech materials than the lexical lookup 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 recognized in spite of a mismatch between the input and that word's canonical representation. In these simulations, however, the mismatch parameter was adjusted during optimization. Even though shortlist was therefore operating with an optimized mismatch parameter, it appears that the DP search algorithm in SpeM works somewhat better in dealing with noncanonical input.

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 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, whereas 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 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 recognize 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 that 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 recognize 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, ten

Bosch, et al. (2003), we argued that this would not have been the case. Cucchiarini, Binnenpoorte, and Goddijn (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 noncanonical, 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, because 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) suggested 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 because the computational analysis (Section 2.1) suggests that an effective prelexical level is essential for large-vocabulary speaker-independent word recognition.

5. Recognition of words in continuous speech

5.1. Temporarily lexically ambiguous input

The simulations reported in Section 4 show that SpeM is able to recognize 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 recognize words in continuous speech is 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 focused 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 recognizer. 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 nonoverlapping 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 that 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 [ʃɪpɪŋkwɑɪəri] into the correct sequence of words.

5.1.1. Method and material

First, the APR component of SpeM was trained on British English. Forty-four acoustic phone models, 1 silence model, and 2 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 recognize *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 optimized for these specific inputs. Instead, we selected the same parameters as were optimized in previous related simulations on *ship inquiry* (simulations in which a linear sequence of phones rather than real speech was used as input; Scharenborg, McQueen, et al., 2003). In addition to the word-activation values generated by SpeM, we also examined the 10 best paths generated by the search module.

5.1.2. Results and discussion

Appendix A shows the 10 best lists for each of the recordings. Fig. 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 recordings). 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,* whereas the segmentation *ship inquiry* can be found at Rank 3 (see Appendix A). For the second recording, the most likely segmentation is *chip inquiry,* whereas 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 acoustic costs associated with the phonemes in *shipping, chip, shook,* and *ship* as calculated by the APR. Toward 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.

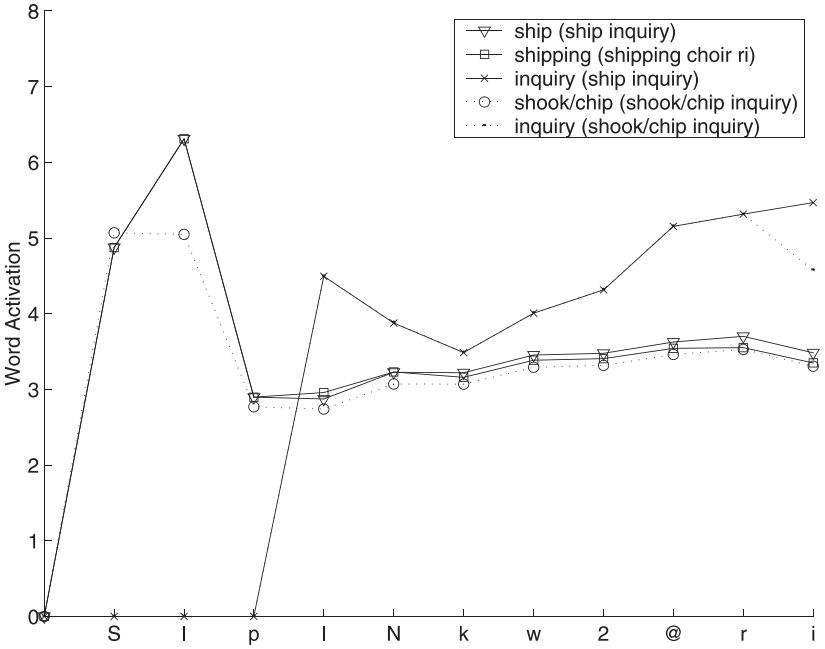

Fig. 6. The mean word activation flows for three recordings of 'ship inquiry'. The y-axis in all panels displays the word activation; the x-axis shows the phonemes in DISC-format (Burnage, 1990) in the input, as they arrive over time.

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, because 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 [ʃɪpɪŋkwɑɪəri] was hardly ever found. The phone sequences found in these lists instead included [ʤɪtɪŋgwɑɪrɪ], [ʤɪtɪŋgwɑɪθrɪ], [ʤɪtɪŋkwʌbrɪ], and [ʤʊdɪŋkwɑɪdrɪ]. Furthermore, there were, on average, 3.03, 4.69, and 3.79 different phonemes (for the first, second, and third recordings, 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 [ʃɪpɪŋkwɑɪəri] 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.

5.2. Lexical competition in spoken-word recognition

In Section 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 fewer 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 4) or on one specific two-word sequence (Section 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?

5.2.1. 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 strong–weak (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 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.

At test, SpeM was asked to recognize the recorded items. The APR module converted the acoustics of each of the recordings into probabilistic phone graphs. Subsequently, these phone

Table 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		
	Stimulus	Target	Embedding Word	Stimulus	Target	Embedding Word
Word onset	do*mes*	*mess*	domestic	*sac*rif	*sack*	sacrifice
Nonword onset	ne*mess*	*mess*	—	*sack*rek	*sack*	—

Note. WS = weak–strong; SW = strong–weak.

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.

5.2.2. 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. Fig. 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 (i.e., of *domestic*); Position 3 is aligned with the start of the target word (i.e., 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 (i.e., of *sack,* and thus also the embedding word, i.e., of *sacrifice*). Note, however, that because the nodes on the *x* axis correspond to the number of nodes in the output graph of the APR, they thus may reflect phones that 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. Fig. 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 in the matched word onset case. The lower panel

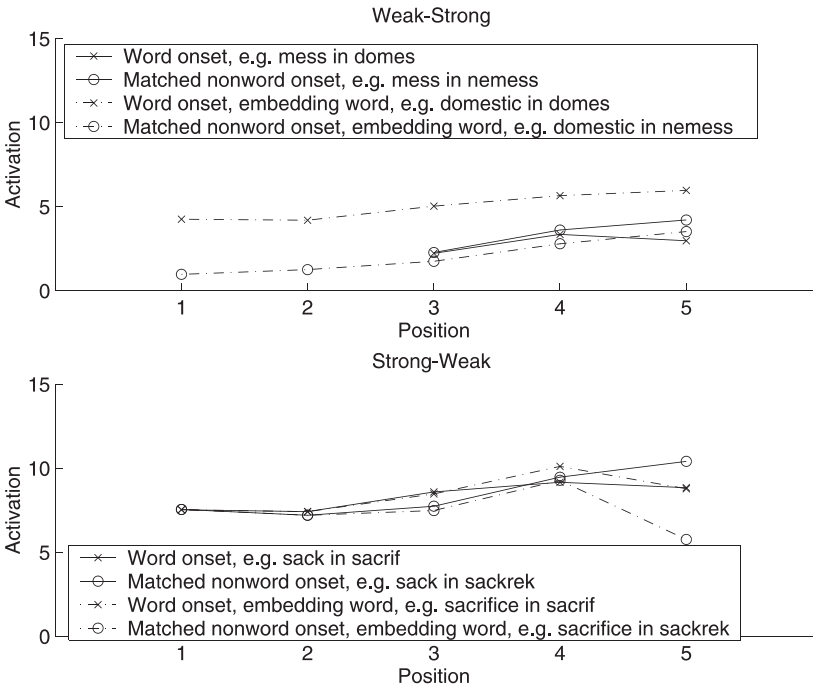

Fig. 7. Mean activation levels for the materials in the domes-simulation for the four stimuli types. In the upper panel, position '1' is aligned with the start of the embedding word (e.g., of 'domestic'); position '3' is aligned with the start of the target word (e.g., of 'mess'). In the lower panel,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

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 non-matched-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.

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 Fig. 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 3–5 in the upper panel of Fig. 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 1–3 in the lower panel of Fig. 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. These 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.

5.3. The PWC 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 favor 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 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 disfavored 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 argued that the PWC helps the speech recognizer 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 that 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 nonexisting) English word (e.g., the syllable *vuff* in *vuffapple*). Can SpeM simulate this result?

5.3.1. 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 gives an example of each of the eight stimulus types. For a full list of the materials, see Norris et al. (1997).

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 Figs. 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.

5.3.2. 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. Fig. 8 shows the activation flows in the control case when the PWC mechanism is disabled; Fig. 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 last/first phoneme of the target word (0). Thus, for targets with preceding context, "+1" is the second segment of the target word, whereas for targets with following context, +1 is the first segment of the context. As was the case for the results plotted in Fig. 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 that overlap partially in time. They do however obey chronological ordering.

Norris et al. (1997) found that words preceded or followed by residues that were possible words were more easily recognized by human subjects (resulting in faster reaction times and fewer errors) than words preceded or followed by residues that were not possible words. Fig. 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, 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. (1997). When the PWC mechanism is enabled, however, SpeM is able to correctly simulate these findings, as is shown in Fig. 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).

Comparing the word-activation flows in Figs. 8 and 9 shows that the activations in Fig. 9 are overall lower than the activations in Fig. 8. This is due to the PWC mechanism. First, the addi-

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

	Monosyllabic Words		Bisyllabic Words	
Residue	Preceding Context	Following Context	Preceding Context	Following Context
Impossible	*f*egg	sea*sh*	*f*apple	sugar*th*
Possible	*maff*egg	sea*shub*	*vuff*apple	sugar*thim*

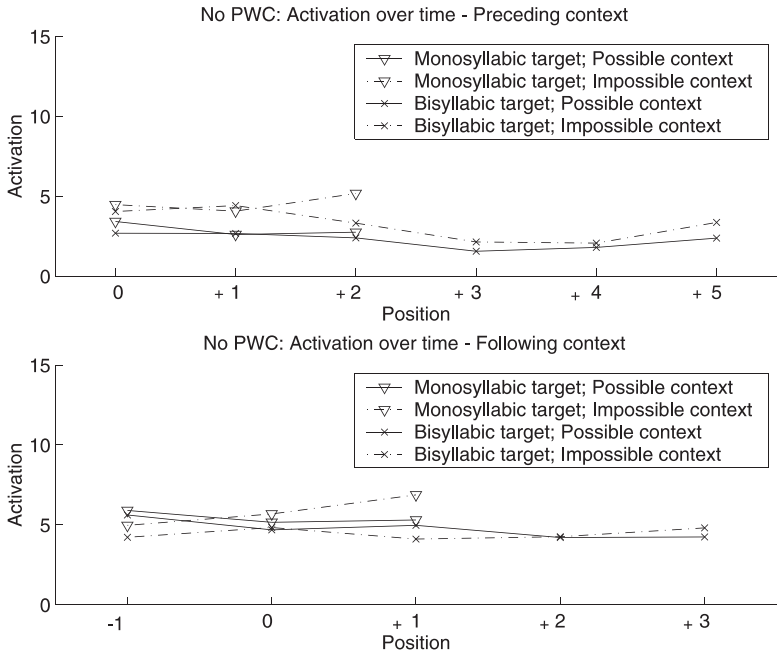

Fig. 8. Mean target activation levels for the materials in the PWC simulation while the PWC mechanism was *dis*abled. The upper panel shows the activation levels for target words with preceding context. The lower panel shows the activation levels for the target words with following context. The activation 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.

tion 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 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.

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

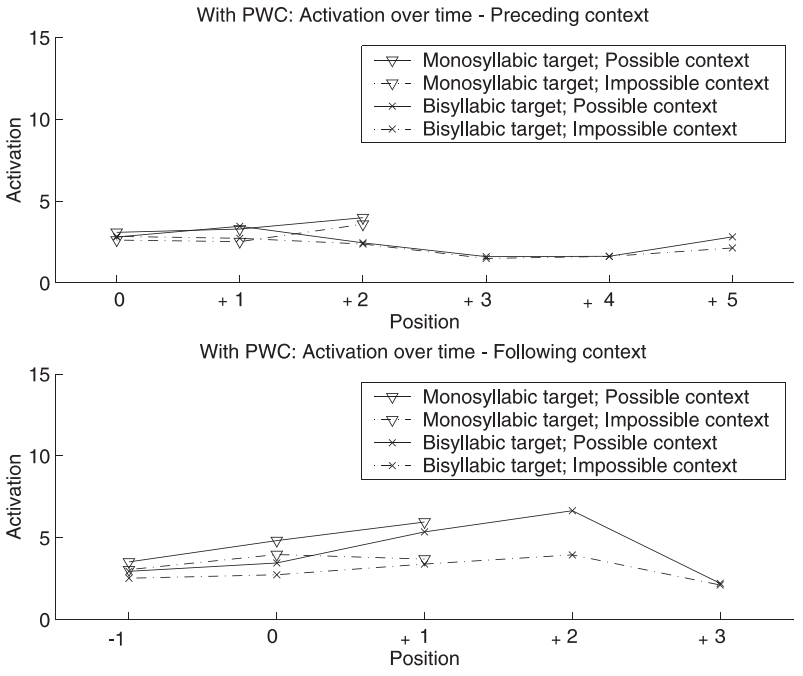

Fig. 9. Mean target activation levels for the materials in the PWC simulation while the PWC mechanism was *enabled*. The upper panel shows the activation levels for target words with preceding context. The lower panel shows the activation levels for the target words with following context. The activation 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.

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 two out of the four cases.

As shown in Table 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 it will also contribute to the pattern

Table 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)

	Monosyllabic Words (%)						Bisyllabic Words (%)					
	Preceding Context			Following Context			Preceding Context			Following Context		
Residue	H	1B	10B	H	1B	10B	H	1B	10B	H	1B	10B
Impossible	52	79	29	39	85	6	18	98	40	38	96	17
Possible	57	77	19	28	69	10	14	90	40	17	92	4

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, because 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 behavior is more deterministic than human word-spotting behavior. Although 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 participant to participant. 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 favor 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 disfavoring the parse *ch apple,* given the input *chapel*). Again, these simulations show that SpeM can be used to simulate performance in specific psycholinguistic experiments.

6. General discussion

In this article, we attempted to bridge the gap that has existed for decades between the research fields of HSR and ASR. According to Marr (1982), every complex information processing system, including any speech recognizer, can be described at three different levels: the computational, the algorithmic, and the implementational. In this article, we offered a computational analysis of speech recognition, with an emphasis on the word recognition process. We focused 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, because 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), which was built using techniques from ASR. SpeM is not just a reimplementation of shortlist; it represents an important advance over existing models of HSR in that it is able to recognize 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 realization of words in the speech signal). SpeM strongly outperformed shortlist in its ability to recognize words from spontaneous speech, spoken by a large number of different talkers in a noisy envi-

ronment, 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, deletion, or substitition of phonemes from those words' canonical pronunciations. The probabilistic prelexical level also allows SpeM to recognize 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 that 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.

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 behavioral studies being simulated, rather than using some idealized form of input representation. These benefits are illustrated by the simulations reported here. First, as in the *ship inquiry* simulations, detailed analyses of handcrafted (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 these 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 modeling to continue to use idealized 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 modeling; 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 modeling 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 modeling 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 cannot recognize 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 recognizing speech, or might even make it fail to recognize speech altogether. ASR modeling has of course been forced to deal with those problems (ASR systems have to be reasonably successful in recognizing words in real-world speech communication situations). The ASR approach adopted in SpeM thus offers a new way of looking at specific modeling 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 recognizer can inform issues of active theoretical debate in psychology. Although models incorporating process interaction (Section 2.6), or episodic recognition (Section 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 online interaction, or to show how it might be possible to build a practical large-vocabulary recognizer based on episodic representations?

In addition, the integrated search procedures used in ASR lead 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 or paths through the search lattice. This is the most obvious way of dealing with sentence context effects in a model such as 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 noninteractive 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 characterized 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 Fig. 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 or words 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 modeling of word frequency effects and with respect to the effects of phonetic mismatch on word recognition. When using Bayes's rule to calculate lexical activation, as in SpeM, there is, for example, no need to have an explicit inhibition mechanism to handle mismatching input such as the [ʃ] in [ʃIgərɛt] (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. (2004) also adopt a Bayesian approach in related work developing the shortlist model. Although the model of Norris et al. (2004) 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. (2004) drove their model from input designed to reflect the characteristics of human prelexical processing. That article discusses the theoretical implications of a Bayesian approach to HSR in more detail than is possible here.

In assessing the value of models such as 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, Stinchcombe, and White (1989) that connectionist networks are universal approximators. That is, algorithms such as Viterbi search could be implemented as connectionist networks. If our analysis is correct, given that the human brain can recognize 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 nonconnectionist models. For more general arguments in favor of explanations at a computational level, rather than in terms of mechanisms or implementations, the reader is referred to Anderson (1990).

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. (2003a) showed that this feature of SpeM allows the model to recognize words before their acoustic offset. Continuous and early recognition measures could be of considerable value in ASR systems, which often do not provide online 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 2, both human and machine word recognizers 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 generalize across both speakers and words. Two-stage ASR systems, similar to the cascaded processing described in Section 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, morphophonological, morphosyntactic, and domain knowledge) into the system (see, e.g., Demuynck 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, because 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 PWC 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.

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 recognizer. 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 recognizer. 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, Allerhand, & Giguere, 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.

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 recognize speech and to approach these questions by focusing on algorithms or implementations. Here, we have presented a computational-level analysis of the task of recognizing 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 emphasized by the implementation of the speech-based model of HSR, 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.

Notes

1. It is well known that episodic models can form abstractions (e.g., Hintzman, 1986). This type of generalization, however, applies to new tokens of categories that have previously been presented to the model (e.g., new tokens of previously presented words), not to novel categories (e.g., previously unencountered words). We are therefore not arguing that episodic models are completely unable to generalize. Nevertheless, they are unable to take advantage of what they have learned about the set of words in their previous experience in recognizing a novel word.
2. A phone is the smallest identifiable unit found in a stream of speech that can be transcribed with an International Phonetic Association symbol. A phoneme is the smallest contrastive phonological unit in the sound system of a language.

Acknowledgments

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.

References

Allopenna, 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: Lawrence Erlbaum Associates, Inc.

Andruski, J. E., Blumstein, S. E., & Burton, M. (1994). The effect of subphonetic differences on lexical access. *Cognition, 52,* 163–187.

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,* 395–408.

Bouwman, G., Boves, L., & Koolwaaij, J., (2000). Weighting phone confidence measures for automatic speech recognition. In *Proceedings of the COST249 Workshop on Voice Operated Telecom Services* (pp. 59–62). Ghent, Belgium.

Burnage, G. (1990). *CELEX: A guide for users.* Nijmegen, The Netherlands: CELEX.

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.

Cucchiarini, C., Binnenpoorte, D, & Goddijn, S. M. A. (2001). Phonetic transcriptions in the Spoken Dutch Corpus: How to combine efficiency and good transcription quality. In *Proceedings of Eurospeech* (pp. 1679–1682). Aalborg, Denmark: Kommunik Grafiske Løsninger A/S.

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.

Demuynck, K., Laureys, T., Van Compernolle, D., & Van hamme, H. (2003). FlaVoR: A flexible architecture for LVCSR. In *Proceedings of Eurospeech* (pp. 1973–1976). Rundle Mall, Australia: Casual Productions.

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: Lawrence Erlbaum Associates, Inc.

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.

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.

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.

Hazen, T. J. (2000). A comparison of novel techniques for rapid speaker adaptation. *Speech Communication, 31,* 15–33.

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. In *Proceedings of the Workshop on Prosody in Automatic Speech Recognition and Understanding* (pp. 61–66). Red Bank, NJ.

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. In *Proceedings of Eurospeech* (pp. 2699–2702). Bonn, Germany: ESCA.

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]. *Proceedings of the IEEE, 88*(8).

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science, 20,* 137–194.

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.

Lesser, V. R., Fennell, R. D., Erman, L. D., & Reddy, D. R. (1975). Organization of the hearsay–II: Speech understanding system. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 23,* 11–23.

Luce, P. A. (1986). A computational analysis of uniqueness points in auditory word recognition. *Perception and Psychophysics, 39,* 155–158.

Luce, P. A., Goldinger, S. D., Auer, E. T., & Vitevitch, M. S. (2000). Phonetic priming, neighborhood activation, and PARSYN. *Perception and Psychophysics, 62,* 615–625.

Luce, P. A., & Pisoni, D. B. (1998). Recognizing spoken words: The neighborhood activation model. *Ear and Hearing, 19,* 1–36.

Maddieson, I. (1984). *Patterns of sounds.* Cambridge, England: 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., et al. (2000). Speech and language technologies for audio indexing and retrieval [Special Issue]. *Proceedings of the IEEE, 88*(8).

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information.* San Francisco: Freeman.

Marslen-Wilson, W., & 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. (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:* In *Psycholinguistic and computational perspectives* (pp. 148–172). Cambridge, MA: MIT Press.

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. (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. (2005). Speech perception. In K. Lamberts & R. Goldstone (Eds.), *The handbook of cognition* (pp. 255–275). London: Sage.

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. (2005). *The mental lexicon is not episodic: A belated reply to Goldinger (1998).* Manuscript in preparation.

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.

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., & 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, The Netherlands: 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.

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. (2005). How do computational models help us develop better theories? In A. Cutler (Ed.), *Twenty-first century psycholinguistics: Four cornerstones* (pp. 331–346). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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. *Behavioral 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. (2004). *Shortlist II: A Bayesian model of continuous speech recognition.* Manuscript in preparation.

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. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 25–28).

Perkell, J. S., & Klatt, D. H. (Eds.). (1986). *Invariance and variability of speech processes.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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.* Upper Saddle River, NJ: Prentice Hall.

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. (2002). Pronunciation variation modelling in a model of human word recognition. In *Proceedings of Workshop on Pronunciation Modeling and Lexicon Adaptation* (pp. 65–70). Estes Park, CO.

Scharenborg, O., McQueen, J. M., ten Bosch, L., & Norris, D. (2003). Modelling human speech recognition using automatic speech recognition paradigms in SpeM. In *Proceedings of Eurospeech* (pp. 2097–2100). Rundle Mall, Australia: Casual Productions.

Scharenborg O., ten Bosch, L., & Boves, L. (2003a). "Early recognition" of words in continuous speech. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop* (pp. 61–66). Rundle Mall, Australia: Casual Productions.

Scharenborg O., ten Bosch, L., & Boves, L. (2003b). Recognising "real-life" speech with SpeM: A speech-based computational model of human speech recognition. In *Proceedings of Eurospeech* (pp. 2285–2288). Rundle Mall, Australia: Casual Productions.

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. *Journal of the Acoustical Society of America, 114*(6), 3032–3035.

Spinelli, E., McQueen, J. M., & Cutler, A. (2003). Processing resyllabified words in French. *Journal of Memory and Language, 48,* 233–254.

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.

Sturm, J., Kamperman, H., Boves, L., & den Os, E. (2000). Impact of speaking style and speaking task on acoustic models. In *Proceedings of ICSLP* (pp. 361–364). Beijing, China: China Military Friendship Publish.

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.

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]. *Proceedings of the IEEE, 88*(8).

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,* 288–298.

Woodland, P. C. (2001). Speaker adaptation for continuous density HMMs: A review. In *Proceedings of the ISCA Workshop on Adaptation Methods for Speech Recognition* (pp. 11–19). Sophia-Antipolis, France.

Zwitserlood, P. (1989). The locus of the effects of sentential-semantic context in spoken-word processing. *Cognition, 32,* 25–64.

Appendix A

The following table displays the 10 best segmentations as calculated by SpeM for the three recordings of *ship inquiry.* The column Segmentation shows the sequence of recognized words in DISC-format (Burnage, 1990). The column Total Cost shows the total path cost as calculated by SpeM (see Section 3.3). The ordering of the paths is done on the basis of the total path cost.

Segmentations as calculated by SpeM for the three recordings of "ship Inquiry"

	Segmentation	Total Cost
First Recording		
1	SUk INkw2@ri	863.890
2	SUk Inkw2@rIN	864.000
3	SIp INkw2@ri	864.080
4	Sut INkw2@ri	864.180
5	SIp Inkw2@rIN	864.190
6	JIp INkw2@ri	864.240
7	SIt INkw2@ri	864.280
8	Sut Inkw2@rIN	864.290
9	JIp Inkw2@rIN	864.350
10	SIt Inkw2@rIN	864.390
Second Recording		
1	JIp INkw2@ri	744.930
2	SIp INkw2@ri	745.320
3	JIt INkw2@ri	745.400
4	JVb INkw2@ri	745.660
5	JIk INkw2@ri	745.700
6	SIt INkw2@ri	745.780
7	SUk INkw2@ri	745.870
8	Sut INkw2@ri	745.920
9	S@d INkw2@ri	745.950
10	Jip INkw2@ri	746.150
Third Recording		
1	SIp INkw2@ri	779.770
2	JIp INkw2@ri	780.200
3	SIt INkw2@ri	780.240
4	SUk INkw2@ri	780.330
5	Sut INkw2@ri	780.370
6	S@d INkw2@ri	780.560
7	JIt INkw2@ri	780.670
8	SIpIN kwQri	780.810
9	SIbin kwQri	781.080
10	SIp Inkw2@ rIt	781.130