Recognising ‘Real-life’ Speech with SpeM: A Speech-based Computational Model of Human Speech Recognition

Odette Scharenborg, Louis ten Bosch, Lou Boves

A^{2}RT, Department of Language and Speech
University of Nijmegen, The Netherlands
{O.Scharenborg,L.tenBosch,L.Boves}@let.kun.nl

Abstract

In this paper, we present a novel computational model of human speech recognition – called SpeM – based on the theory underlying Shortlist. We will show that SpeM, in combination with an automatic phone recogniser (APR), is able to simulate the human speech recognition process from the acoustic signal to the ultimate recognition of words. This joint model takes an acoustic speech file as input and calculates the activation flows of candidate words on the basis of the degree of fit of the candidate words with the input.

Experiments showed that SpeM outperforms Shortlist on the recognition of ‘real-life’ input. Furthermore, SpeM performs only slightly worse than an off-the-shelf full-blown automatic speech recogniser in which all words are equally probable, while it provides a transparent computationally elegant paradigm for modelling word activations in human word recognition.

1. Introduction

In this paper, we introduce a novel computational model for explaining key effects of human word recognition. The existing computational models of human speech recognition (HSR) usually model only parts of the human speech recognition process. One of the parts that virtually all HSR models lack is a module that converts the acoustic speech signal into a segmental representation of this signal. Instead, HSR models assume an error-free symbolic representation of the speech signal as input. In [1], we attempted to produce an *end-to-end* speech-based model of HSR, using a joint model of an automatic phone recogniser (APR) and an existing computational model of HSR, viz. Shortlist [2]. As a (partial) model of HSR, Shortlist has a very successful track record in modelling a wide range of results from psycholinguistic experiments related to (human) word recognition.

This joint model works as follows. Based on an input speech signal, the APR generates a single linear phone string. Shortlist takes this sequence as input and generates a sorted list of candidate words with their activations based on the degree of fit between the candidate word and the input. The word (sequence) with the highest activation is considered as the recognised word (sequence). (For more information on Shortlist, the reader is referred to [2]).

The APR and Shortlist individually have a good record in explaining a large number of effects on their domain. However, word recognition experiments showed that a joint model consisting of an APR and Shortlist does not necessarily yield an end-to-end model of HSR that is able to deal with ‘real-life’ input adequately [3]. Apparently, one cannot take for granted that a combination of the best models of two research sides yields the best overall model. Perhaps this is not too surprising, since neither system was designed with the intention of being interfaced with the other.

The central limitation of the joint APR-Shortlist model is that it makes ‘hard’ decisions, both at the level of the output of the APR (which must generate the input phone string for the Shortlist model), and in the goodness-of-fit metric used in the word search process implemented in Shortlist. The form of the input of Shortlist – a single string of phones – implies that, if the APR is to be directly connected to Shortlist, the APR is forced to make hard decisions about the representation of the speech signal based on the acoustic information only. Secondly, the search of Shortlist matches input phone strings to the phone strings stored in the internal Shortlist lexicon in a way that makes no allowance for insertions or deletions; the matching process is intolerant for deviations from the number of phones in the canonical form of words as stored in the internal Shortlist lexicon. Since in spontaneous speech many substitutions, deletions, and insertions occur, this is a quite unrealistic assumption.

In this paper, we present a new computational model of HSR called SpeM (SPEech based Model) without the above-described limitations. SpeM and Shortlist have the same underlying theoretical basis, but the implementation is very different. While Shortlist is implemented using a neural network, the word search module within SpeM is implemented using a dynamic programming (DP) technique. By doing so, deviations from the canonical representations can be dealt with more naturally and elegantly, and we expect that SpeM will be able to deal with real-life speech input more adequately. Furthermore, instead of using a single linear input phone string – as is required for Shortlist – the input of SpeM can be a probabilistic phone graph. In this way, both hard constraints are relaxed.

The research question investigated in this paper is whether SpeM is indeed a better recogniser of real-life speech than Shortlist. Furthermore, SpeM’s performance will be compared with the performance of an off-the-shelf automatic speech recognition system (ASR) on the same task. In the following sections, we will discuss SpeM in more detail. Furthermore, experiments will be described that investigate the performance of SpeM, Shortlist, and the ASR on the task of recognising real-life speech. The paper ends with a discussion of the results of the experiments and a short conclusion.

2. SpeM

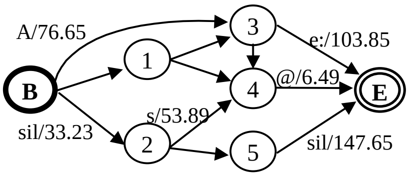
2.1. Lexicon and input representations

For the word search process, SpeM uses a lexicon containing the words it should be able to recognise and a phone graph.

Internally, the lexicon (a list of Dutch city names, see Section 3.3) is represented as a lexical tree in which the entries (words) share common prefix phone strings (this is called a cohort). Each path through the tree represents a full pronunciation variant of a word. In SpeM, pronunciation variants of words are treated as separate words. Henceforth, 'word' can also mean 'pronunciation variant'. The tree has one root node (denoted 'B') and as many end nodes as there are entries (i.e. pronunciation variants) in the lexicon.

The input is a phone graph generated by the APR, more specifically; it is an a-cyclic directed connected graph, with one start node (denoted 'B') and one end node. Figure 1 shows a graphical representation of an input phone graph. Each arc (connection between two nodes) carries a phone and its bottom-up evidence in the acoustic signal (acoustic cost). For the sake of clarity, not all phones and their acoustic costs are shown.

Figure 1. A graphical representation of an input phone graph.



2.2. Search algorithm

The search for the best matching (sequence of) word(s) is the search for the cheapest path through the product graph defined by the product of the lexical tree and the input phone graph. The path must start at node (B, B). The search is implemented using DP. The *total cost* of a path in the product graph is defined as the sum of the acoustic cost, the phone matching cost, the Possible Word Constraint (PWC) cost (for more information on the PWC, see [4]), the history, and the word entrance penalty in which

- *Acoustic cost:* this cost is the negative log likelihood as calculated by the APR.
- *Phone matching cost:* this cost is associated with the cost of the current transition. There are four possibilities:
  - Identical: no costs.
  - Substitution, deletion, and insertion: with their associated costs. For each, the associated cost is tuned by hand.
- *PWC cost:* this cost is related to whether a (series of) phone insertion(s) occurring between the word and an utterance boundary is phonotactically well formed (and thus pertains to a possible word) or not.
- *History:* this cost is inherited from the mother node – it is the cost of the cheapest path to the mother node.
- *Word entrance penalty:* cost to start a new word.

The implemented search algorithm is time-synchronous and breadth-first. During the dynamic search, many search space nodes – related to probable but also improbable and duplicate paths – will be created. To save computational costs, a pruning mechanism has been implemented. Three types of pruning are used to keep the number of 'active' nodes manageable [5]:

- *Number of nodes:* A maximum number of search space nodes are kept in memory. After each cycle of creating new nodes, the active nodes are sorted according to their total cost; only the top maximum number of nodes is kept, the rest is discarded.
- *Local score pruning:* A new space node is only created if the total cost of the new path is less than the total cost of the best path up to that point plus a pre-set value.
- *No duplicate paths:* Of the search space nodes that represent duplicate paths, only the node with the cheapest path is kept.

It is possible that an end node of the lexical tree is reached before the end node of the input graph. In that case, the search through the lexical tree starts from the root node again. In this way, sequences of words can be recognised.

The algorithm has five well-defined parameters that have a direct correspondence to HSR modelling effects. These are the substitution, deletion, insertion, and PWC cost and the word entrance penalty.

2.3. Output

After generating all search space nodes that belong to an input phone node, and after pruning the worst and redundant paths, the best 'N' paths up to that point are available. Each path consists of the word hypotheses and their activation, and the total cost of the path. If a path refers to a word that has not yet reached the leaf node in the lexical tree, the intermediate printed output will mark this cohort with an asterisk (*). An example of the output of SpeM after presenting the spoken word *hilversum* (a Dutch city name) at the input is shown in Table 1. Here, the value of 'N' is six.

Table 1. An example of the output of SpeM after presenting 'hilversum' as input to the joint model.

Path	**Activation**	**Total cost**
hILv@RsYm	2.52e-06	1065.84
prIns@m*	9.06e-07	1066.14
wInsYm	4.95e-07	1075.47
hILv@RsY*	1.36e-06	1076.72
prIns@*	4.86e-07	1077.32
xILz@	2.67e-07	1086.12

The path consisting of the single word *hilversum* has the highest activation and lowest total cost. The absence of an asterisk indicates that at this particular point in the input graph, an end leaf in the lexical tree is reached.

The second best candidate word is /prIns@m/. The asterisk behind the candidate word indicates that the path has not yet reached the leaf node in the lexicon: it is a cohort shared by for instance the Dutch city name *prinsenbeek*.

The cohort /hILv@RsY/ (row four) has a higher total cost than the full word /hILv@RsYm/, because it has an /m/-deletion at the end of the word. Therefore, an extra deletion penalty is added to the total score of the path.

3. Experiments

3.1. Introduction

As previously explained in the introduction, SpeM is a computational model of human speech recognition. In [6], we show that the model is indeed able to correctly simulate the results found in psycholinguistic studies. On top of this result, in this paper, we will show to what extent SpeM is better in recognising real-life speech input than Shortlist is. To that end, we conducted a series of word recognition experiments in which a conventional HMM-based ASR system, a joint model of the APR and Shortlist (henceforth called APR+Shortlist), and a joint model of the APR and SpeM (henceforth called APR+SpeM) all perform the same task. The experiments were devised in such a way that all results are optimally comparable with respect to lexicon size and language model.

3.2. The speech recognition systems

3.2.1. The APR

The APR is based on the Phicos automatic speech recognition system [7]. For the APR, we trained 36 context-independent HMM phone models, one silence model, one model for hesitations such as 'uh', and one noise model. Each phone, hesitation, and noise model has a linear left-to-right topology with three pairs of two identical states, one of which can be skipped. The silence model consists of one state. The APR is based on a phone loop with optional silence between each phone pair and optional start and trailing silence guided by a phone bigram. The APR parameters were kept the same across all experiments.

3.2.2. The ASR system

The ASR is also based on the Phicos automatic speech recognition system. 37 context-independent phone models, one noise, one silence, and one garbage model were trained. Each phone and noise model has a linear left-to-right topology with three pairs of two identical states, one of which can be skipped. The silence and garbage model each consist of one state. In [8], this ASR is described in more detail.

Since there is no knowledge on the frequency of words in either Shortlist or SpeM, for a fair comparison a language model has been used for the ASR system in which all words are equally probable.

3.3. Data

3.3.1. Training and test material

For training and testing the APR and the ASR, data from the Dutch Directory Assistance Corpus (DDAC) were used [9]. The material to train the acoustic models comprises 24,559 utterances (DDAC-train). Most utterances consist of either a Dutch city name or 'ik weet het niet' ('I don't know') pronounced in isolation, although in a few cases audible hesitations like 'uh' were present.

The independent test set (DDAC-test) consists of a selection of 10,506 utterances (not overlapping with DDAC-train) with a total number of 11,517 words. These utterances may also contain disfluencies and connected speech responses like 'haarlem noordholland' (i.e., a city name plus the name of a province). All utterances were recorded over a fixed telephone line.

3.3.2. Lexicons

In the first experiment, the lexicon consists of 924 entries: city names, Dutch province names and 'ik weet het niet' ('I don't know'). For each entry in the lexicon, a unique canonical phonemic representation was available. This lexicon will be referred to as the 'Base' lexicon in the remainder of this paper.

In spontaneous speech, many substitutions, deletions and insertions occur. Consequently, the number of actually produced phones may differ from the number of phones in the standard transcription. The psycholinguistic theory underlying Shortlist makes no claim about the manner in which humans cope with pronunciation variation. Specifically, there is nothing in the theory that promotes the exclusive use of citation forms in the mental lexicon. To avoid unnecessary problems due to pronunciation variation we decided to add pronunciation variants to the Base lexicon. This enriched lexicon – called 'Pron' – contains on average 2.6 pronunciation variants per word, and has 2,428 entries. More details on how the pronunciation variants were created can be found in [1].

3.4. Experimental set-up

The aim of the experiment is to investigate the performance of the three different recognition systems.

The utterances of DDAC-test were used as input to the three systems. In the case of the ASR, the signals were presented to the ASR and at the output, words were recognised. In the case of APR+SpeM and APR+Shortlist, the APR was used to create a segmental representation of the acoustic signal. In APR+Shortlist, this segmental representation consists of a single linear phone transcription of the acoustic signal; in APR+SpeM, this segmental representation is a probabilistic phone graph. For this particular test set, the average number of arcs in a phone graph is 140, while the number of nodes in a phone graph varies between 18 and 112.

The performance is calculated in terms of accuracy: the percentage of utterances for which the correct word was recognised. A word is correctly recognised if it has the highest activation – in the case of APR+Shortlist and APR+SpeM – or is the first best – in the case of the ASR – and is identical to the word in the orthographic transcription of DDAC-test.

Table 2: Results on the DDAC-test utterances for APR+Shortlist, APR+SpeM, and the ASR system.

Model	APR+Shortlist		APR+SpeM		ASR	
Lexicon	Base	Pron	Base	Pron	Base	Pron
Acc (%)	31.7	54.2	72.3	77.2	84.9	84.5

3.5. Results

Table 2 shows the results for the three systems. Next to the accuracy ('Acc') of the models, the type of lexicon is shown. Table 2 clearly shows that APR+SpeM outperforms APR+Shortlist on the task of recognising real-life speech. In the case of the Base lexicon, APR+SpeM's performance is more than twice as high as APR+Shortlist's.

With respect to the comparison between the performance of APR+SpeM and the ASR, we see that APR+SpeM comes remarkably close to the ASR results. The difference in performance is only 7.3% in the Pron lexicon condition.

The results show that using pronunciation variants improves the performance of both APR+SpeM and APR+Shortlist. However, the gain for APR+Shortlist is by far the largest. The word search in Shortlist is – as already indicated – intolerant for deviations in number of phones from the canonical form of words as stored in the internal Shortlist lexicon. For most words, the pronunciation variants mostly contain fewer phones than the canonical form of the word; these pronunciation variants resemble the input more increasing the matching process of Shortlist. The gain of adding pronunciation variants for SpeM is less apparent, since the DP implementation of the search in itself is better able to deal with insertions and deletions.

In the case of the ASR, adding pronunciation variants has no effect – in fact, there is a small degradation in performance when using the Pron lexicon. The finding that the benefit of adding pronunciation variants is only slight in ASR systems is already confirmed by many other experiments (see e.g. [10]).

4. Discussion

There is a clear difference in performance between APR+Shortlist and APR+SpeM. Since the theory underlying Shortlist and SpeM is identical, it is fair to say that the difference in performance is caused by the different implementations of SpeM and Shortlist. The two major differences between the two models are related to 1) the search algorithm and 2) 'soft' vs. 'hard' decisions. In principle, the relative contributions of the DP search and the relaxation of the requirement that the input of Shortlist must consist of a one-dimensional phone string can be established by repeating the experiments with SpeM with a string instead of a graph as input. However, such an experiment is not expected to enhance our understanding of the theory underlying Shortlist.

The DP technique used for the implementation of the search module in SpeM is far better able to deal with input containing a different number of phones than candidate words in the internal lexicon than the search mechanism implemented in Shortlist. This is clearly shown by the figures in Table 2.

In SpeM, a probabilistic phone graph is used as input instead of a linear phone string. Despite its internal richness, the phone graph created by the APR remains a rather crude representation of the speech signal. We hypothesise this to be the reason for the difference in performance for APR+SpeM and the ASR. Some degradation can also be expected, since SpeM is based on a relatively simple transparent dynamic programming technique, while the search in ASR relies on a more sophisticated and well-engineered code base. A crucial advantage of SpeM is that the word search is explicitly focussed on the treatment of cohorts, which allows SpeM to produce output hypotheses before the end of a (long) word is reached. Therefore, this model can be used as a tool for studying the results found in experiments of word classification with final-gated signals in HSR studies.

5. Conclusions

In this paper, we presented a novel computational model of human speech recognition – called SpeM – based on the theory underlying Shortlist. SpeM is able to correctly simulate the results found in many psycholinguistic studies.

A series of word recognition experiments showed that SpeM is far better able in dealing with real-life input than the original implementation of Shortlist. Furthermore, the joint model of the APR and SpeM performs only slightly worse than an off-the-shelf full-blown automatic speech recogniser in which all words are equally probable, while it provides a transparent computationally elegant paradigm for modelling word activations of human word recognition.

6. Acknowledgements

Part of this work was carried out while the first author was visiting the Medical Research Council Cognition and Brain Sciences Unit, Cambridge, UK. The authors would like to thank Dennis Norris and Gies Bouwman for their help on implementing SpeM.

7. References

[1] Scharenborg, O., Boves L., "Pronunciation Variation Modelling in a Model of Human Word Recognition," *Proceedings of the Workshop on Pronunciation Modeling and Lexicon Adaptation*, Estes Park, CO, USA, pp. 65-70, 2002.

[2] Norris, D., "Shortlist: a Connectionist Model of Continuous Speech Recognition," *Cognition 52*, 189-234, 1994.

[3] Scharenborg, O., Boves L., de Veth, J., "ASR in a Human Word Recognition Model: Generating Phonemic Input for Shortlist," *Proceedings of ICSLP*, pp. 633-636, 2002.

[4] Norris, D., McQueen, J.M., Cutler, A., Butterfield, S., "The Possible-Word Constraint in the Segmentation of Continuous Speech," *Cognitive Psychology 34*, 191-243, 1997.

[5] Ney, H. and Aubert, X., "Dynamic Programming Search: from Digit Strings to Large Vocabulary Word Graphs," In: *Automatic Speech and Speaker Recognition, (C.-H. Lee et al, eds.) Kluwer Academic Publishers, Boston*, pp. 385-413, 1996.

[6] Scharenborg, O., McQueen, J.M., ten Bosch, L., Norris, D., "Modelling Human Speech Recognition using Automatic Speech Recognition Paradigms in SpeM," To appear in *Proceedings of Eurospeech*, 2003.

[7] Steinbiss, V., Ney, H., Haeb-Umbach, R., Tran, B.-H., Essen, U., Kneser, R., Oerder, M., Meier, H.-G., Aubert, X., Dugast, C., Geller, D., "The Philips Research System for Large-vocabulary Continuous-speech Recognition," *Proceedings of Eurospeech*, pp. 2125-2128, 1993.

[8] Bouwman, G., Boves, L., "Using Information on Lexical Stress for Utterance Verification," *Proceedings of the Workshop on Prosody in ASRU*, Red Bank, NJ, USA, pp. 29-34, 2001.

[9] Sturm, J., Kamperman, H., Boves, L., den Os, E., "Impact of Speaking Style and Speaking Task on Acoustic Models," *Proceedings of ICSLP*, pp. 361-364, 2000.

[10] Kessens, J., Wester, M., Strik, H., "Improving the Performance of a Dutch CSR by Modeling Within-word and Cross-word Pronunciation Variation", *Speech Communication 29*, 193-207, 1999.