Modelling Fine-phonetic Detail in a Computational Model of Word Recognition

Odette Scharenborg

Centre for Language and Speech Technology, Radboud University Nijmegen, The Netherlands

O.Scharenborg@let.ru.nl

Abstract

There is now considerable evidence that fine-grained acoustic-phonetic detail in the speech signal helps listeners to segment a speech signal into syllables and words. In this paper, we compare two computational models of word recognition on their ability to capture and use this fine-phonetic detail during speech recognition. One model, SpeM, is phoneme-based, whereas the other, newly developed Fine-Tracker, is based on articulatory features. Simulations dealt with modelling the ability of listeners to distinguish short words (e.g., 'ham') from the longer words in which they are embedded (e.g., 'hamster'). The simulations with Fine-Tracker showed that it was, like human listeners, able to distinguish between short words from the longer words in which they are embedded. This suggests that it is possible to extract this fine-phonetic detail from the speech signal and use it during word recognition.

Index Terms: fine-phonetic detail, word recognition, computational modelling

1. Introduction

In normal everyday communication, listeners appear to be able to recognise the intended word sequences almost effortlessly. Even in the case of fully embedded words such as 'ham' in 'hamster', listeners can make the distinction between the two interpretations even before the end of the first syllable 'ham'. There is now considerable evidence from psycholinguistic and phonetic research that sub-segmental (i.e., subtle, fine-grained, acoustic-phonetic) and supra-segmental (i.e. prosodic) detail in the speech signal modulates human speech recognition, and helps the listener segment a speech signal into syllables and words (e.g., [1],[2],[3]).

It is this kind of information that appears to help the human perceptual system distinguish short words from the longer words in which they are embedded. For instance, it is shown that the lexical interpretation of an embedded sequence is related to its duration [3]; a longer sequence tends to be interpreted as a monosyllabic word more often than a shorter one. These results seem to question the validity of the phoneme as the unit of recognition in human speech recognition.

In this paper, we investigate whether 'fine-phonetic detail' [4] or 'non-segmental information' in the speech signal can actually be extracted from the speech signal and used during word recognition. We do so by testing two computational models in modelling the human ability to detect and use these non-segmental cues during speech recognition. Both computational models of human word recognition are based on the theory underlying the Shortlist model [5]; they however differ in the unit of recognition they use: phonemes for SpeM [6] and articulatory features (AFs) for the newly developed model Fine-Tracker [7]. Both computational models are built using techniques from the field of automatic speech recognition (ASR), making them able to recognise actual speech signals. One clear advantage of this is that these models can be tested with precisely the same stimulus materials as used in the behavioural studies being simulated, instead of using some idealised form of input representation as is done by most other computational models of human word recognition.

If successful, these experiments will provide support for the theory that non-segmental cues in the speech signal modulate speech recognition. Additionally, it will provide evidence to help answer the question about the ideal unit of recognition in speech recognition.

2. The computational models

2.1. SpeM

SpeM (Speech-based Model of human word recognition, [6]) is an extended implementation of Shortlist, which has proven successful in simulating parts of the human word recognition process, while using real speech as input. The theory underlying SpeM and Shortlist claims that the speech recognition process consists of two levels: the prelexical level, at which the incoming acoustic signal is mapped onto prelexical representations, and the lexical level, at which these representations are mapped onto lexical representations. Following Shortlist, the prelexical representations in SpeM take the form of phones.

SpeM consists of two modules, one for each level. The first module is an automatic phone recogniser (APR), which creates probabilistic phone lattices. The APR is based on HTK [8]: it uses 37 monophone models, each consisting of 3 emitting states, which were trained on the read speech part of the Spoken Dutch Corpus (SDC) [9].

The second module is a word search module, which parses the phone lattices in order to find the most likely (sequence of) words, and computes for each word its activation based on the accumulated acoustic evidence. The search module finds the sequence of words with the smallest distance between the sequence of phones on the path through the phone lattice and the phonemic representations of the words in the lexicon using a time-synchronous and breadth-first DP algorithm. Each phone insertion, deletion, and substitution is penalised according to parameter values which can be tuned separately. The output of SpeM consists of an N-best list of hypothesised parses. Each parse contains words, word-initial cohorts (words sharing phone prefixes), silence, and any combination of these, except that a word-initial cohort can only occur as the last element in the parse.

2.2. Fine-Tracker

Like SpeM and Shortlist, Fine-Tracker consists of two levels/modules: a module that creates a prelexical representation of the speech signal and a word search module.

Table 1. Specification of the AFs, their AF types, and the number of hidden nodes in the MLPs.

AF	AF type	#hidden nodes
manner	plosive, fricative, nasal, glide, liquid, vowel, sil	300
place	bilabial, labiodental, alveolar, (pre)palatal, velar, glottal, nil, sil	200
voice	+voice, -voice	100
fr-back	front, central, back, nil	200
round	+round, -round, nil	100
height	high, mid, low, nil	200
dur-diph	long, short, diphthong, nil	200

The biggest difference between SpeM and Fine-Tracker is the form of the prelexical representations. Fine-Tracker is specifically designed to 'track' fine-phonetic detail in the speech signal. It uses articulatory features, which are abstract classes characterising the articulatory properties of speech sounds in a quantised form [10]. Table 1 shows an overview of the AFs used by Fine-Tracker. Note that *fr(ont)-back*, *round*, *height* and *dur(ation)-diph(thong)* only apply to vowels.

The first module creates a multi-dimensional feature vector for every 10 ms of speech of the speech signal. Each feature vector has a continuous value between 0 and 1 for each of the AF types in Table 1, resulting in 32-dimensional feature vectors. The value of each AF type can be regarded as a measure of activation of this AF type, which can thus be traced over time.

In the current version of Fine-Tracker, the first module is implemented as multi-layer perceptrons (MLPs). For each of the AFs, one MLP was trained using the NICO Toolkit [11] on 4000 randomly selected utterances from the read speech part of the SDC [9]. Each MLP consisted of three layers. The input layers had 39 nodes. The hidden layers had hyperbolic tan transfer functions and a different number of nodes depending upon the AF. The optimal number of hidden units was determined through tuning experiments and is listed in the third column of Table 1. The output layer was configured to estimate the posterior probability of the AF value given the input. The number of output nodes is identical to the number of AF values (see Table 1). When training each MLP, the performance on a validation set (consisting of a similar set of utterances as used for the simulations described here (Exp 1B from [3])) was monitored and training was terminated when the validation set's error rate began to increase.

In the Fine-Tracker lexicon, the words are also represented in terms of multi-dimensional feature vectors. Because the values of the AF types can take any value between 0 and 1, speech phenomena such as coarticulation, assimilation, and nasalisation of vowels can easily be encoded through feature spreading. Essential in Fine-Tracker is the fact that the number of feature vectors per phoneme can be set for each phoneme or word separately. The word search module of Fine-Tracker is able to deal with these subtle differences in lexical representations.

The word search module compares the multi-dimensional feature vectors with the candidate words in the lexicon in order to find the most likely (sequence of) words, and it computes the activation flows of these candidate words. It does so by determining the sequence of words with the smallest distance through the search space spanned by the multi-dimensional input feature vectors and the lexical feature representations of the words. The search algorithm is time-synchronous and breadth-first and uses a many-to-one mapping, since multiple 10ms feature vectors need to be mapped onto a single lexical feature vector. For each path through the search space the *total cost* is calculated, which consists of the sum of the:

- Word entrance penalty: cost to start a new word.
- Step-in-lexicon: a penalty associated with making a 'step' in the lexicon, but not in the input. This results in a lexical feature vector being inserted (similar to a phone insertion).
- Step-in-input: a penalty associated with making a 'step' in the input but not in the lexicon.
- Word not finished penalty: at the end of the input, all cohorts that do not correspond to words get a penalty.
- History: this cost is inherited from the 'mother' node – it is the cost of the cheapest path to the mother node.
- Distance measure: currently, the averaged squared distance. The relative weight of the distance measure and the penalties above is determined by a distance weight parameter. There is an option in Fine-Tracker to implement other distance calculation measures.

As in SpeM, only the most likely candidate words and paths are considered; therefore several pruning mechanisms (see [12], for an overview) have been implemented:

- Number of nodes: the maximum number of hypotheses kept in memory during the word search.
- 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 the local score pruning value.
- No duplicate paths: of identical word sequences, only the cheapest path is kept.

All parameters can be tuned separately. The output of the search module consists of an N-best list of hypothesised parses containing words, word-initial cohorts, silence, and any combination of these, with the restriction that a word-initial cohort can only occur as the last element in the parse.

The Fine-Tracker software is implemented in JAVA and is distributed under the GNU General Public License (GPL) via [7]. It runs on any platform where Java Runtime Environment version 1.6 or newer is available.

3. Experimental set-up

3.1. Experiment by Salverda et al.

For the simulations, we use the same stimulus materials as in the eye-tracking studies reported in [3]. In those experiments, participants listened to sentences and were asked to click on the object (one out of four pictures, presented on a computer screen) mentioned in the sentence. This 'target' word is a multi-syllabic word of which the first syllable also constitutes a monosyllabic word (e.g. 'hamster' and the embedded monosyllabic word 'ham'). In total, 28 Dutch target words in 28 utterances were used.

The target words were created in two ways[1]: 1) the first syllable of the target word is replaced through cross-splicing by a different recording of the first syllable of the multi-syllabic target word (referred to as MULTI); 2) the first syllable is replaced by a recording of the monosyllabic embedded word (referred to as MONO). An example:

[1] Actually, three different forms of the target words were contrasted in [3], but in the simulations reported here we only use two, i.e. those used in Exp 1A in [3], therefore only these two sets of stimuli are described here.

Original	Zij dacht dat die ham*ster*$_a$ *verdwenen was*
	*Zij dacht dat die ham*ster$_b$ verdwenen was
	(She thought that that hamster had disappeared)
	Zij dacht dat die ham$_c$ stukgesneden was
	(She thought that that ham had been sliced)
Cross-spliced	1. Zij dacht dat die ham$_b$ster$_a$ verdwenen was
	2. Zij dacht dat die ham$_c$ster$_a$ verdwenen was

During the experiment, participants' eye movements were monitored. Analysis of the eye movements showed that there were significantly more transitory fixations to pictures representing monosyllabic words (e.g., 'ham') if the first syllable of the target word (e.g., 'hamster') had been replaced by a recording of the monosyllabic word than when it came from a different recording of the first syllable of that target word.

3.2. Model testing and predictions

For the simulations, the speech files are parameterised with 12 MFCC coefficients and log energy and augmented with first and second temporal derivatives resulting in a 39-dimensional feature vector. These feature vectors are used as input to the APR module used by SpeM and the MLP module (the number of input nodes of the MLP is equal to the dimensionality of the MFCCs) used by Fine-Tracker. The lexicon used by both models consisted of 28,402 entries.

The activations of both the target words (e.g. 'hamster') and the monosyllabic embedded words ('ham') are extracted from the N-best lists computed by the models, and plotted over time. In our comparison of the models' word activation plots with the human results, we consider the amount of transitory fixations as a degree of the word activation during human word recognition. This means we can directly compare the output of the two models with the results of the listeners.

In the MONO condition, we expect the activation of the embedded word to be higher than the word activation of the target word, since the first syllable of the target word was taken from a recording of the monosyllabic word. Reversely, we expect the word activation of the target word to be higher than that of the embedded word in the MULTI condition, since here the first syllable of the target word came from a different recording of the target word.

SpeM first maps the acoustic signal onto phoneme-like prelexical representations, which are subsequently mapped onto lexical representations consisting of phoneme strings. Since words such as 'ham' and 'hamster' have the exact same first three phonemes, we expect no difference in the word activations of the target words and the embedded words in the fine-phonetic detail simulations. We therefore predict that SpeM will not be able to correctly simulate the human results. However, Fine-Tracker uses feature vectors of AF features and different lexical representations for the embedded and target words. It is thus in theory able to use duration and thus distinguish between 'ham' and 'hamster'. We expect differences in word activation between the target words cross-spliced with a recording of the monosyllabic embedded word and with a different recording of the same word.

4. Results and Discussion

4.1. SpeM

The lexicon used during the simulations consists of only one pronunciation variant per word, i.e. the canonical phoneme representation. Of the 28 target words, SpeM was able to correctly recognise 17 target words, of which 12 were recognised correctly for both sets of stimuli, using a 50-best list. Figure 1 shows the averaged word activations over time of the target ('o') and the embedded words ('▽') for the MONO (solid line) and MULTI (dashed line) conditions. The words are aligned such that the phoneme position indicated with '0' is the start of the second syllable.

Comparing the MONO and the MULTI conditions shows that the word activations in the MONO conditions are higher than in the MULTI conditions. However, as predicted, there is no difference in word activations for the target and embedded words in either condition. This means that SpeM is indeed not able to correctly simulate the effects found in human speech recognition.

4.2. Fine-Tracker

For the current simulation, the lexical feature vectors were obtained by substituting all phonemes in the SpeM lexicon with their canonical AF values. The number of lexical feature vectors for each phoneme in a word was determined by hand, although ideally this should be determined automatically. Setting this number by hand, however, will give a good idea of what is maximally possible if the automatic method would do it correctly. Since phoneme duration decreases with increasing number of syllables (e.g., [13]; this was also found to be true for the acoustic data used in this study – on average 245 ms in the MULTI condition, and 265 ms in MONO condition, see [3]), the number of AF feature vectors per phoneme in monosyllabic words was one more than the number of AF feature vectors per phoneme for the first syllable of the target words.

Fine-Tracker was able to correctly recognise 14 target and 13 embedded words in the MONO condition, and eight target (of which seven were recognised in both conditions) words and nine embedded words in the MULTI condition, using a 50-best list. Figure 2 shows the averaged word activations over time of the target ('o') and the embedded words ('▽') for the MONO (solid line) and MULTI (dashed line) conditions. Again, the words are aligned such that the phoneme position indicated with '0' is the start of the second syllable.

There is a clear difference in activations for the target and embedded words. Like for the human listeners, in the MONO condition, the embedded words clearly have a higher word activation than the target words. Although this result is not reversed in the MULTI condition, the difference in word activation between the embedded and target words is far smaller than in the MONO condition. These results show that Fine-Tracker is able to detect and use fine-phonetic detail during speech recognition.

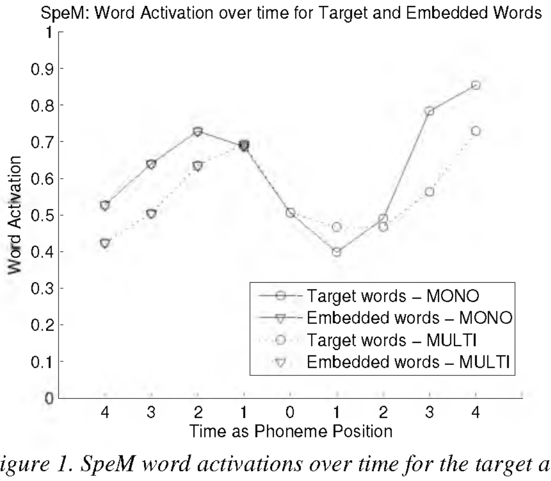

Figure 1. SpeM word activations over time for the target and embedded words in the MONO and MULTI conditions.

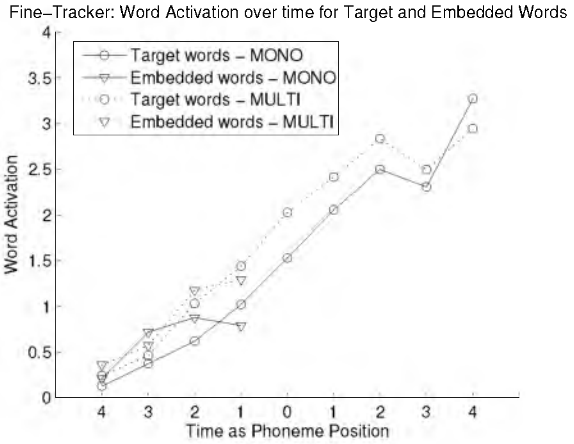

Figure 2. Fine-Tracker activations over time for the target and embedded words in the MONO and MULTI conditions.

5. Concluding remarks

We compared two computational models of word recognition on their ability to capture and use fine-phonetic detail during speech recognition. Simulations dealt with modelling the ability of listeners to distinguish short words (e.g., 'ham') from the longer words in which they are embedded (e.g., 'hamster') using the same acoustic material as was used for the behavioural study presented in [3].

The first modelling results obtained with Fine-Tracker are promising. Follow-up research will focus on improving its recognition and modelling performance by implementing new distance measures and improving the lexical representations.

As predicted, the phoneme-based model SpeM was not able to distinguish between two words with the same phoneme sequence. It, thus, did not correctly model the human results. One could try and build an APR which might be able to distinguish between longer and shorter versions of the phonemes, and distinguish syllables in monosyllabic and multi-syllabic words in the lexicon using these 'new' phonemes. However, the distributions of the length of phonemes in monosyllabic and multi-syllabic overlap considerably. It is, therefore doubtful whether this approach will work. The SpeM results challenge the validity of the phoneme as the unit of recognition in human listeners.

Fine-Tracker was able to track fine-phonetic detail in the speech signal. This is due to its unit of recognition in combination with its capability of dealing with subtle differences in lexical representations. Like human listeners, it showed a preference (i.e., a higher word activation) for the embedded mono-syllabic word when the first syllable of the target word came from a recording of the monosyllabic word. Although this result was not reversed when the target word was cross-spliced with the first syllable of another recording of the target word, the word activation difference of the embedded and the target words was smaller. These results show that Fine-Tracker is sensitive to fine-phonetic detail and that it can extract and use it during speech recognition. These results strengthen the theory that non-segmental cues in the speech signal modulate speech recognition.

The advance of articulatory feature approaches of speech recognition is the ability to model pronunciation variation through simple feature spreading. It thus provides a flexible alternative to the standard phoneme-based or 'beads-on-a-string' paradigm in ASR [14]. However, two questions remain to be answered through further research. First, how should the AF-based lexicons be trained to go beyond the currently used canonical feature vectors? Second, how can the optimal number of feature vector per phoneme be determined automatically?

6. Acknowledgements

This research was supported by a Veni-grant from the Netherlands Organisation for Scientific Research. The author would like to thank A.P. Salverda (Univ. of Rochester) for kindly providing the acoustic data, A. Hämäläinen for providing the acoustic models for the SpeM simulations, F. Kusters for implementing Fine-Tracker, L. ten Bosch for help with the development of Fine-Tracker. Also, thanks to L. ten Bosch and L. Boves for providing useful comments on an earlier version of this paper, and to A. Cutler (MPI for Psycholinguistics) for providing Fine-Tracker's name.

7. References

[1] Davis, M.H., Marslen-Wilson, W.D., Gaskell, M.G., "Leading up the lexical garden-path: Segmentation and ambiguity in spoken word recognition", J. Exp. Psych.: H Perception and Performance, 28, 218-244, 2002.

[2] Kemps, R., Ernestus, M., Schreuder, R., Baayen, R.H., "Prosodic cues for morphological complexity: The case of Dutch plural nouns", Mem.&Cogn. 33, 430-446, 2005.

[3] Salverda, A.P., Dahan, D., McQueen, J.M., "The role of prosodic boundaries in the resolution of lexical embedding in speech comprehension", Cogn. 90, 51-89, 2003.

[4] Hawkins, S., "Roles and representations of systematic fine phonetic detail in speech understanding", J. Phonetics, 31, 373-405, 2003.

[5] Norris, D., "Shortlist: A connectionist model of continuous speech recognition", Cogn. 52, 189-234, 1994.

[6] Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., "How should a speech recognizer work?" Cognitive Science, 29 (6), 867-918, 2005.

[7] http://www.finetracker.org

[8] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., "The HTK book (version 3.2)", Tech. Report, Cambridge Univ., Engineering Dept., 2002.

[9] Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M., Baayen, H., "Experiences from the Spoken Dutch Corpus project", Proc. of LREC, Las Palmas, Gran Canaria, p. 340-347, 2002.

[10] Kirchhoff, K., "Robust speech recognition using articulatory information", Ph.D. thesis, Univ. of Bielefield, 1999.

[11] Ström, N., "Phoneme probability estimation with dynamic sparsely connected artificial neural networks", The Free Speech Journal, 5, 1997.

[12] Ney, H., Aubert, X., "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, 1996.

[13] Nooteboom, S.G., "Production and perception of vowel duration in Dutch: A study of durational properties of vowels in Dutch", PhD thesis, Univ. of Utrecht, The Netherlands, 1972.

[14] Ostendorf, M., "Moving beyond the 'beads-on-a-string' model of speech", Proc. of IEEE ASRU Workshop, Keystone, CO, pp. 79-84, 1999.