ASR Decoding in a Computational Model of Human Word Recognition

Louis ten Bosch, Odette Scharenborg

CLST, Radboud University Nijmegen, The Netherlands
{L.tenBosch,O.Scharenborg}@let.ru.nl

Abstract

This paper investigates the interaction between acoustic scores and symbolic mismatch penalties in multi-pass speech decoding techniques that are based on the creation of a segment graph followed by a lexical search. The interaction between acoustic and symbolic mismatches determines to a large extent the structure of the search space of these multi-pass approaches. The background of this study is a recently developed computational model of human word recognition, called SpeM. SpeM is able to simulate human word recognition data and is built as a multi-pass speech decoder. Here, we focus on unravelling the structure of the search space that is used in SpeM and similar decoding strategies. Finally, we elaborate on the close relation between distances in this search space, and distance measures in search spaces that are based on a combination of acoustic and phonetic features.

1. Introduction

Both the research areas of automatic speech recognition (ASR) and human speech recognition (HSR) investigate the recognition process from the acoustic signal to a sequence of recognised units. For ASR, the target is to automatically transcribe the speech signal in terms of a sequence of items as close as possible to a reference transcription (e.g., [1], [2]). In HSR, the focus is on understanding how human listeners recognise spoken utterances. Based on HSR experiments, theories about specific parts of the HSR system are refined. To put the theories to further test, they are implemented in the form of computational models for the simulation and explanation of HSR (e.g., Shortlist, [3], Trace, [4]).

Recently, a computational model of human word recognition has been developed that makes use of techniques developed in the area of ASR [5]. The model, called SpeM, provides a successful and concrete demonstration of the computational parallels between HSR and ASR, by making the links between HSR and ASR as explicit as possible. SpeM decodes speech based on the theory underlying Shortlist; its implementation, however, is entirely different (see section 2).

SpeM works as a multi-pass decoder. A phone graph that is produced in the first pass is input for a lexical search in the second pass. The ability of SpeM to simulate data from human word recognition experiments is ultimately based on the structure of its search space. In multi-pass speech decoding, this search space is determined by the interaction between acoustic scores of segments on the one hand, and penalties for symbolic mismatches (phone insertions, deletions and substitutions) on the other. In order to better understand the mechanisms underlying SpeM-like decoding, and its potential usefulness for ASR we will investigate the interaction between acoustic and symbolic mismatches in more detail. Finally, we will show that the decoding can be linked to approaches in ASR that use phonetic features in combination with acoustic features. In order to introduce these issues, we first give a brief overview of the SpeM model.

2. SpeM

The SpeM model is implemented as a multi-pass decoder (see Figure 1).

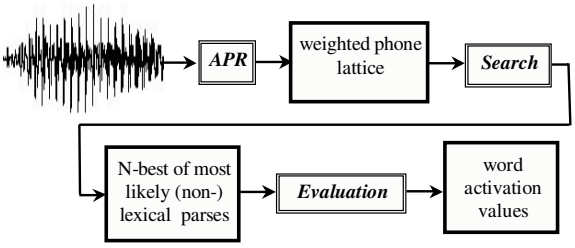

Figure 1. An overview of the implementation of the SpeM model (figure adapted from [5]).

In the first pass, an automatic phone recogniser (APR) processes the input speech signal and generates a (weighted) phone lattice. This lattice provides a probabilistic phone representation of the speech signal, and is input for the second pass which deals with the lexical search. Because the phone lattice is ultimately interpreted via the search algorithm, we will first pay attention to the search algorithm itself, before we discuss the search space (which is spanned by the phone lattice and the lexicon) in more detail in the next section.

The SpeM search module performs a search for sequences of lexical items such that the phonemic representation of these sequences (as determined by a vocabulary) is optimally matching the phone sequences in the lattice. The search is a node-synchronous Viterbi-like forward pass through a graph that is the product of the phone graph and the lexical graph (tree). This product graph is the actual search space. Each arc π in the product graph corresponds to an arc $\alpha(\pi)$ in the phone graph and an arc $\beta(\pi)$ in the lexical graph, and has a weight equal to the sum of the weights of $\alpha(\pi)$ and $\beta(\pi)$. The weight of $\alpha(\pi)$ is the acoustic score calculated by the APR; the weight of $\beta(\pi)$ consists of the unigram and bigram language model (LM) scores. Unlikely hypotheses are pruned away.

A 'garbage' phone model is included in the lexicon, which can be mapped onto phones that do not belong to a lexical item. The search is able to deal with symbolic mismatches between phone sequences in the phone graph

and the lexicon, by allowing (symbolic) insertions, deletions, and substitutions. Each type of mismatch has its own penalty, which can be tuned independently. Thanks to this flexibility, each parse may therefore consist of lexical items, word-initial cohorts (words sharing phone prefixes), non-lexical items, silence, and any combination of these (except that a word-initial cohort can only occur as the last element in the parse). The output of the search is an *N*-best list of hypothesised parses, with the (acoustic and LM) costs.

The last module of SpeM performs the evaluation. In this module, the *N*-best list of parses is processed to generate, for each hypothesised word, a 'word activation' measure that varies over time. Since the word activation measure and its potential for use in ASR have been described elsewhere ([5], [7]), the evaluation module will not be further discussed here.

3. The search space

As indicated above, the APR creates a weighted phone graph as a phonetic probabilistic representation of the acoustic signal. From the perspective of the search following the APR, an important issue is to what extent the phone graph must capture the phonetic detail in the signal in order to maximise the likelihood of containing phone sequences that are easy to map to lexical solutions. The basic assumption is that the APR is able to produce a phone lattice that is an accurate phonetic representation of the speech signal, including locally phonetically plausible variations, without being guided or constrained by lexical information.

The search space in a multi-pass decoder as described here is the product graph of the phone graph and the lexical graph. While looking for optimal paths through this search space, the precise decisions of the search algorithm depend on the interaction between acoustic scores (from the phone lattice) and symbolic mismatch scores (handled by the search mechanism). Therefore, in order to understand the mechanisms underlying the search in multi-pass decoders, we need to investigate the interaction between these scores and to relate this interaction to the eventual goal of the search for lexical solutions in the phone graph.

Several factors determine the structure of the phone graph, and thereby the chance that a phone sequence related to a sequence of lexical items can be found along a path through this graph. Apart from evident factors such as the quality of the acoustic models and (implementation) details concerning splitting and recombination of arcs during the phone search, three factors have a decisive impact on the structure and contents of the resulting phone lattice: a) the phone insertion penalty; b) the beam width during the phone search by the APR; c) the use (and weighting) of a phone *N*-gram during the phone search. The details of the search in the second pass depend on the global characteristics of the graph. For example, one would expect a close –but potentially complex – relation between the phone insertion penalty in the APR and the insertion and deletion penalties in the search.

The interaction between acoustic scores and symbolic mismatches is fully determined by their balance during the search. For the search module to be able to find a lexical sequence with associated phone sequence P_c, there must be a phone sequence on a path Q through the phone lattice with the following property:

$$P_c = \text{argmin}_P\{\min_Q(score(Q) + d(P,Q) + LM(P))\} \qquad (1)$$

This expression is the mathematical formulation of the forward pass in the search. The term *score(Q)* is a shorthand for –log(P(X|Q)). The signal X is given, P is the hypothesised lexical path, and Q is a path variable, running over the set of all paths available in the phone lattice. The term *score*(Q) denotes the total path score of Q as defined by the phone lattice, while d(P, Q) denotes the sum of all penalties for symbolic mismatches between the phone sequences P and Q. The final term LM(P) denotes the language model score of the word sequence associated with P. Evidently, the minimising path Q depends on the hypothesised P.

Eq. 1 implies that the penalties for symbolic insertions, deletions, and substitutions are not free model parameters, but instead must be closely related to the distribution of acoustic path scores in the lattice. For example, let P denote a specific (arbitrary) hypothesis, and assume that for some path Q the term d(P, Q) is made up by I insertions, D deletions, and S substitutions. In that case, the application of Eq. 1 has the very same effect as the evaluation of the score of the canonical path P in a *new* lattice L' that is obtained from the original lattice L by expanding *all* possible paths in L by allowing exactly I insertions, D deletions, and S substitutions with their corresponding costs. This new lattice L' (which does not physically exist, but is virtually constructed and explored during the search) depends on I, D, and S and, by construction, contains the sequence P. Repeating the same argument for any hypothesis P, it follows that the *eventual* search space where *all* canonical sequences can be found is effectively the *union* of virtual lattices L'(I, D, S) such that I, D, S >= 0. As a consequence, the entire distribution of the path scores in this union lattice is the union of the original score distribution H and shifted copies of this distribution: {H, H+*cost*(I), H + 2**cost*(I),…, H+*cost*(I) + *cost*(D), …, H + *cost*(D), …, H + *cost*(S), …}. And the only thing that really counts in the search is how 'far' in this union lattice any canonical phone sequences are alive. Since the structure of the union lattice is fully determined by L and by the symbolic mismatch costs, this means that the penalties for substitution, insertions, and deletions must be considered in relation to the structure of L, in particular to the distribution of the acoustic scores of paths in L that are canonical or almost canonical (i.e., with a small number of mismatches).

It is therefore of importance to know more about the score distribution of the phone lattice itself. To that end, we have examined a set of phone lattices from 669 utterances with read speech, selected from the Spoken Dutch Corpus (CGN, [8]). The phone lattices have been created using the HTK recogniser with acoustic monophone 3-state left-to-right HMMs with 8 gaussians/state, and a phone zerogram. The values for the phone insertion penalty and the beam width have been chosen such that the resulting phone lattices are phonetically plausible. First, the phone insertion penalty was adjusted such that the resulting average number of phones in the best path was equal to the number of phones in

the canonical phone transcription defined by the orthographic transcription and the vocabulary (the resulting average number of phones per second is about 13). Second, the beam width has been adjusted such that the time-averaged number of arcs with *different* phone labels is close to 3, i.e. a plausible number of realistic phonetic alternatives.

Given these choices, it appears that the number of *arcs* crossing a certain moment is on average 12 (minimum 2, maximum 48). The high number of local options implies that the number of paths through the lattice might be huge. Indeed, the relation between the number N of paths in the lattice and the duration L of the utterance can be approximated by

$$^{10}log(N) = C * L \quad (2)$$

with C equal to about 5.5. The equation implies that an utterance of 2 seconds may have a phone graph with as many as 100 billion paths. The constant C depends on the beam width and on the phone insertion penalty: the larger the beam and the lower the insertion penalty, the larger C will be. This implies that for utterance of a few seconds, even reasonably long N-best lists of phone sequences (of, say, length 50,000) capture only a negligible fraction of the information in the graph. An N-best list is interesting because it captures local information about probabilistic segmentation (e.g., [9]), but it has hardly any relevance for capturing the canonical sequence (actually, the probability that the *complete* graph contains the canonical correct phone sequence decreases rapidly with the length of the utterance, and is for our data set smaller than 1 percent for utterances longer then 1.5 sec).

Much more relevant for the structure of the search space in SpeM-like decoding is the minimum number of substitutions, insertions, and deletions required to construct the canonical sequence from a path through the phone graph, because this gives the 'distance' between the graph and the canonical phone sequence. Table I shows this number (the minimum Levenshtein distance) as a function of the utterance duration for the 669 utterances. The first column refers to the duration category, the second column presents the total number of utterances per category, while the third column contains the Levenshtein distance between the given canonical sequence and its best-matching path through the lattice, averaged over all utterances in the category. (This best-matching path minimises the Levenshtein distance with the canonical phone sequence – it is not necessarily the path with the highest acoustic score.) The fourth column presents the average location of this best-matching path in the phone graph, expressed in percentiles of the entire score distribution of acoustic scores of the paths in the phone graph. '0' means the cheapest path, '10' means at the 10th percentile, etc.

According to table I (last column) the phone graph could be made much smaller, the top 10-15 percent would have been sufficient for this data set. By comparing the Levenshtein distance and the duration, we conclude that the canonical path is about two repairs per second away from its best-matching solution in the graph. Given that the canonical path contains 13 phones/sec, on average 2 out of 13 phones must be 'repaired'.

Table I. The minimum Levenshtein distance and the location of the path that minimised the Levenshtein distance as a function of the utterance duration.

Duration cat. (sec)	*#utt*	*Average Levenshtein distance*	*Location of found path (percentile)*
0.50-0.75	6	1.2-1.4	<5
0.75-1.0	19	2.2	<7
1.0-1.5	54	2.5	<6
1.5-2.0	73	3.3	<4
2.0-3.0	150	3.6-4.2	<6
> 3.0	367	> 4.1	-

Figure 1 shows in another way how the information in N-best lists is only of marginal value for finding complete sequences. It shows the number of different phone sequences among the 5,000-best as a function of the duration of the 669 utterances. Silence arcs have been discarded. As expected, for longer utterances, all phone hypotheses in the 5,000-best list tend to be unique. The 'hockey stick effect' for low durations is due to the fact that short utterances relatively contain more silence than longer utterances which evidently reduces the number of different phone paths.

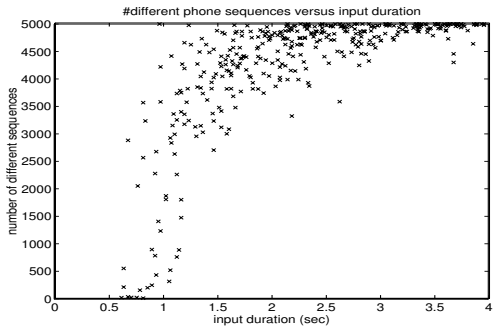

Figure 1. This figure shows, for 669 utterances, the number of *different* phone paths in the 5,000-best list as a function of the duration of the utterance.

3.1. Adding Phonetic Features

The final observation that we want to make is about an interpretation of Eq. 1 that enables us to establish a close link with phone decoding strategies that are based on acoustic feature representations augmented with phonetic features. Minimising the right-hand side of Eq. 1 can be thought of as looking for a path $Q = \{q1, q2\ldots\}$ in such a way that it optimally matches X (by minimising $-log(P(X|q))$) and at the same time minimises $d(P,Q)$. The resulting alignments between the speech frames $\{x1, x2, \ldots\}$, the phones in Q $\{q1, q2, \ldots\}$, and in the canonical phone sequence P $\{p1, p2, \ldots\}$ are schematically shown in Figure 2 (top). However, $d(P, Q)$ is a sum of local symbolic distances between $\{p\}$ and $\{q\}$, a sum which can be represented by the sum of distances between symbolic phonetic feature vectors. Furthermore, the alignment between X and Q implicitly assigns to each frame in X a phonetic representation

inherited from the phones $\{q\}$. So we can rewrite the sum of *score*(Q) and $d(P, Q)$ in Eq. 1 as one *single* distance between two *augmented* sequences: one sequence *augX* of feature vectors $\{x\}$ augmented (via the alignment) with phonetic features from $\{q\}$, and a sequence *augP* of $\{p\}$ augmented with their own phonetic features (Fig. 2, bottom displays the new situation).

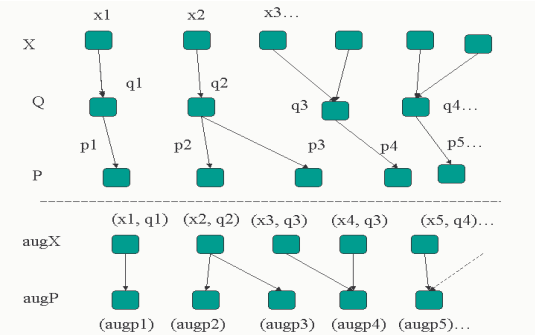

Figure 2. Top: Association between speech frames $\{x\}$, phone sequence $\{q\}$ and $\{p\}$ by alignment via Eq.1. *Bottom*: The same association, with one single distance between augmented representations.

This implies that the search for lexical parses in the phone lattice can be interpreted as a search for a match between an augmented representation of the frames in X and an augmented representation of the segments in P. The correspondence is not always exact, since in Eq. 1, the minimising Q is dependent on P, while here it is assumed that each frame in the speech signal can be assigned a static phonetic feature representation. But we know from other research (e.g., [10], [11]) that such a feature assignment can be done with reasonable plausibility. Furthermore, although the number of different paths in the phone lattice may be large, the *local* variations are mostly within one phonetic class. This means that speech recognition approaches based on combinations of acoustic and phonetic information in the search can be linked in a natural way with a SpeM-like speech decoding. It also shows how the *symbolic* penalties and *acoustic* scores can be brought into one framework.

4. Conclusions

The search problem in SpeM that combines bottom-up acoustic scores with penalties for symbolic mismatches has been studied by considering the interaction between the distribution of acoustic scores and operations on symbols in the phone lattice.. The search space can be regarded as the union of the original phone lattice and virtual lattices that are related to symbolic insertions, deletions, and substitutions. The penalties for symbolic mismatches are closely related to the distribution of the acoustic scores of (near-)canonical paths in the lattice. Phone lattices built with a phone loop with zerogram phone-LM and plausible values for beam width and phone insertion penalty show that the probability of observing the 'correct' phone sequences (i.e., the sequence that is identical to the canonical phone sequence according to the lexicon) decreases rapidly with the length of the utterance. In order to be able to find the correct lexical solution, the flexibility to deal with the symbolic mismatches is absolutely essential. Given the canonical correct phone path, the path through the lattice that minimised the Levenshtein distance has always been found in the top 7 percent of all paths, and the required minimum number of repairs (substitutions or insertions or deletions) was found to be about 2 per second. A SpeM-like search with an acceptable proportion of search errors appears to be feasible in ASR applications.

Finally, we have indicated the close resemblance between the lexical search in SpeM on the one hand, and the approaches in ASR using phonetic features on the other. This relation opens possibilities to integrate the acoustic/phonetic approaches in ASR and the computational modelling of human speech recognition in a more unified paradigm.

5. Acknowledgements

The first author participated in the FP5 project COMIC (nr. IST-2001-32311). Annika Hämäläinen provided the CGN data and the HTK acoustic models. Lou Boves is gratefully acknowledged for helpful comments on previous versions of the text.

6. References

[1] Rabiner, L., Juang, B.-H., "Fundamentals of speech processing". New Jersey: Prentice Hall, 1993.

[2] Jelinek, F., "Statistical methods for speech recognition". Cambridge, MA: MIT Press, 1997.

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

[4] McClelland, J.L., Elman, J.L., "The TRACE model of speech perception", *Cognitive Psychology*, *18*, 1-86, 1986.

[5] Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., "How should a speech recognizer work?", *Accepted for publication in Cognitive Science.*

[6] McQueen, J.M., Speech perception, In K. Lamberts, R. Goldstone (Eds.), *The handbook of cognition* (pp. 255-275). London: Sage Publications, 2004.

[7] Scharenborg, O., ten Bosch, L., Boves, L., "'Early Recognition' of Words in Continuous Speech", *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, US Virgin Islands (CDROM), 2003.

[8] Oostdijk, N., "The design of the Spoken Dutch Corpus". In Peters, P., Collins, P., Smith A. (Eds) *New Frontiers of Corpus Research* (pp. 105-112). Amsterdam: Rodopi, 2002.

[9] Lee, S., Glass, J. "Real-time probabilistic segmentation for segment-based speech recognition", *Proc. ICSLP*, Sydney, Australia. pp. 1803-1806, 1998.

[10] King, S., and Taylor, P. "Detection of phonological features in continuous speech using neural networks", *Computer Speech and Language, 14(4)*, pp. 333-353, 2000

[11] Livescu, K., Glass, J., "Feature-based pronunciation modeling with trainable asynchrony probabilities." *Proc. ICSLP*, Jeju, South Korea, October 2004 (CDROM).