

# Robust Phone Lattice Decoding

Kris Demuyne, Dirk Van Compernelle, Hugo Van hamme

Katholieke Universiteit Leuven – dept. ESAT  
Kasteelpark Arenberg 10, B-3001 Leuven

{kris.demuyne, dirk.vancompernelle, hugo.vanhamme}@esat.kuleuven.be

## Abstract

Most ASR systems adopt an all-in-one approach: acoustic model, lexicon and language model are all applied simultaneously, thus forming a single large search space. This way, both lexicon and language model help in constraining the search at an early stage which greatly improves its efficiency. However, such close integration comes at a cost: all resources must be kept simple. Achieving higher accuracy in unconstrained LVCSR tasks will require more complex resources while at the same time the ‘unconstrainedness’ of the task reduces the effectiveness of the all-in-one approach. Therefore, we propose a modular two-layered architecture. First, a pure acoustic-phonemic search generates a dense phone network. Next a robust decoder finds those words from the lexicon that match well with the phone sequences encoded in the phone network. In this paper we investigate the properties the robust word decoder must have and we propose an efficient search algorithm.

**Index Terms:** speech recognition, LVCSR, phone lattice, search.

## 1. Introduction

Over the years, most HMM based automatic speech recognition (ASR) systems have adopted an all-in-one approach: acoustic model, lexicon and language model (LM) are all applied simultaneously, thus forming a single large search space. This search space is either constructed dynamically [1], statically [2] or mixed static/dynamic [3]. Close integration introduces the task-dependent constraints (lexicon and language model) at an early stage in the search, and hence greatly improves the efficiency of the search. This is especially true for constrained tasks such as small vocabulary tasks or tasks using finite state grammars. However, close integration also comes at a cost. To allow the combination of all knowledge components into a single search space, they must be kept extremely simple. This has particularly inhibited progress at the linguistic level. Consequently, almost all recognizers employ non-optimal linguistic components such as static lexica (lexicalization of morphological processes) and N-gram LM’s.

We believe that, in order to meet the aims set by current research programs –i.e. recognition of unconstrained speech input, higher accuracy, less domain dependency and richer transcription output [4]– more sophisticated linguistic models are indispensable. At the same time the ‘unconstrainedness’ of these new tasks reduces the effectiveness of the all-in-one approach. In [5] we proposed a novel modular two-layered framework called FLaVoR (Flexible Large Vocabulary Recognition). The key aspect of the proposed framework consists of splitting up the search engine into two separate layers. The first layer performs phone recognition and outputs a dense phone network, which acts as an interface to the second layer. In the second layer, the actual word decoding is accomplished by means of a robust decoder. The robust word

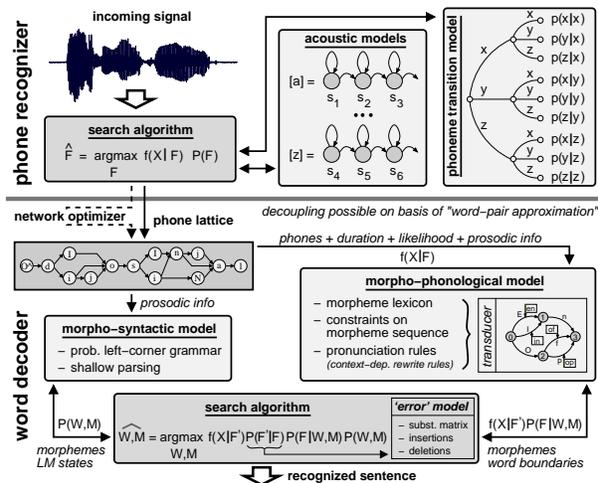


Figure 1: FLaVoR architecture in detail

decoder searches for those word sequences drawn from the lexicon that match well, albeit not always exactly, with the phone sequences encoded in the phone lattice and are considered likely by the language model.

In [5] we focused on the rationale of the FLaVoR-approach, the differences with existing multi-pass strategies and on the phone decoder. The focus in this paper will be on the robust word decoder. A major requirement for this decoder is robustness w.r.t. the mismatch between observed acoustics (encoded in the phone network) and the ideal (e.g. canonical) forms found in the lexicon.

This paper is organized as follows. Section 2 describes the FLaVoR-approach in more detail. The next section handles the experiments. Section 3.1 introduces the task and gives reference results using an all-in-one decoder. In section 3.2, the configuration of the phone decoder (first layer) is presented. The different configurations for the word decoder (second layer) are presented and evaluated in sections 3.3 and 3.4. We conclude the experiment section with a discussion. In section 4 we propose an efficient implementation of the robust search. We end with conclusions.

## 2. The FLaVoR architecture

In this section, we will briefly recapitulate the FLaVoR-architecture. For more details, the rationale behind FLaVoR and differences with existing multi-pass strategies, we refer to [5].

Figure 1 depicts the FLaVoR architecture in more detail. In the first layer, a phone decoder determines the network of most probable phone strings  $F$  given the acoustic features  $X$  of the incoming

signal. The knowledge sources employed are an acoustic model  $p(X|F)$  and a phone transition model  $p(F)$ . The resulting phone network can be augmented with meta-data (prosody, speaker identity, etc.) in order to provide rich information to the second layer.

Important to note is the isolation of the low-level acoustic-phonemic search from the the higher layers by means of a dense phone network. This decoupling is made possible by the high quality of current acoustic modeling and by an extension of the so-called word-pair approximation [6]. For a formal description of this extension we refer to [5]. The decoupling of acoustic and word decoding significantly lowers the event rate. First, there is a pure reduction in data rate, from the 100 feature vectors per second at the input to an average of 12 phones (plus alternatives) per second at the output. In addition, the number of parallel options each input stands for is reduced: while a monolithic search engine has to match all incoming feature vectors with all possible combinations of (context dependent) phones and end positions at that point in the search, the phone network only contains the set of best matching phones with their optimal start and end times. This opens up the possibility of using more complex linguistic components in the word decoder. Another important aspect is the generic nature of the first layer for a full natural language. That is, the phone recognizer can function in any knowledge domain for a specific language. In addition, the phone information itself could be used in certain applications (e.g. language learning) or for handling specific problems (e.g. recognition of proper names).

The phone network and associated meta-data serve as input to the second layer which performs the actual word decoding. The primary knowledge sources used in this pass are the lexicon and the language model, or more generally, a morpho-phonological and a morpho-syntactic component. The morpho-phonological component converts the phone network into sequences of morphemes. The morpho-phonological knowledge consists of a morpheme lexicon, constraints on morpheme sequences and pronunciation rules. All knowledge sources will be combined into one finite state transducer (FST). As was illustrated in the work at AT&T [2] and by ourselves [3], such transducers are a very compact and efficient solution for decoding. The pronunciation rules qualitatively and quantitatively describe the contextual influence on the pronunciation of a sequence of phonemes. More precisely, probabilistic context-dependent rewrite rules formalize the processes of assimilation, insertion, etc. on the intra- as well as the inter-word level. As such these rules provide the indispensable link between the isolated standard phonemic transcription of morphemes as found in the morpheme lexicon and their realization in ‘real-life speech’. The morpho-syntactic language model provides a probability measure for each hypothesized word based on morphological and syntactic information of the word and its context.

The search algorithm links input and knowledge sources together. Next to being efficient, this decoder must also be robust w.r.t. the mismatch between observed acoustics (encoded in the phone network) and the transcriptions provided by the morpho-phonological component. Since only regular pronunciation alternations can be described in the morpho-phonological component, it is up to the search to cope with non-regular pronunciation variants such as dialectal influence, swallowed sounds in fast pronunciation and slips of the tongue. Furthermore, the word decoder must also be robust w.r.t. errors made by the phone recognizer.

Although the need for a ‘robust’ decoder complicates the system, it should not be considered a weakness. In classical decoders, all deviations between the canonical transcriptions in the lexicon and the observed acoustics must be modeled by the acoustic mod-

setup	feb89	oct89	feb91	sep92	mean
	all-in-one decoder				
reference	2.26	2.83	2.13	5.08	3.08
	FLaVoR phone decoder				
phone error rate	8.84	10.2	8.89	12.4	10.1
	FLaVoR word decoder				
baseline	4.88	5.25	3.95	8.05	5.54
baseline + filler	3.75	4.58	3.58	7.97	4.98
filler + rules	3.55	4.17	3.30	7.97	4.75
ins/del + rules	2.77	3.65	2.90	5.47	3.70
single error + rules	2.30	2.87	2.58	4.81	3.14

Table 1: Error rates on the RM test sets using the *XL* phone lattices

els, leading to ‘contaminated’ models. The FLaVoR-approach isolates these phenomena and describes them explicitly, and hence more correctly, by means of rules (regular processes) and a substitution matrix (irregular processes). As is reported in [7] there is evidence that human speech recognition works in similar ways.

### 3. Experiments

#### 3.1. Task & reference results

The Resource Management (RM) task, being a constrained task (word pair grammar) with a limited vocabulary (991 words), is not the best match for the FLaVoR-architecture which aims at unconstrained large vocabulary speech recognition (LVCSR). However, RM is a compact well defined task with few tuning parameters<sup>1</sup> and hence is ideal for the development of new techniques.

For all experiments, we used our in-house state-of-the-art speech recognition system [8]. For the acoustic models we used our default shared gaussian approach, i.e. the density function for each of the 791 cross-word context-dependent tied states is modelled as a mixture over an arbitrary subset of gaussians drawn from a global pool of 7487 gaussians. The mixtures use on average 104.7 gaussians to model the 36 dimensional observation vector. The 36 dimensions were obtained by means of a mutual information based discriminant linear transformation [8] on 24 MEL spectra and their first and second order time derivatives. The word error rates (WER = ins. + del. + sub.) obtained with the acoustic models using our existing all-in-one decoder are given in table 1.

#### 3.2. Layer1: phone decoding

In order to evaluate the two layer FLaVoR-approach, a properly configured phone decoder is needed. A phone trigram was estimated from an automatically generated reference phone transcription of the train database. To create these reference transcriptions, we converted the orthographic transcription of each sentence into a phone network by means of a pronouncing dictionary and a limited set of assimilation rules, and used the Viterbi algorithm to decide on the best path through the network given the acoustic signal. The configuration parameters of the phone decoder (weighting of the phone trigram w.r.t. the acoustic likelihoods) were determined using the feb89 test set. Hence, we choose the weighting that minimizes the differences (phone error rates) between the output of the phone recognizer and the reference transcription. The error rates obtained with the phone decoder are given in table 1.

<sup>1</sup>next to the parameters that control the search effort, only a single ‘word-startup-cost’ is used to balance the impact of the acoustic model and the word pair grammar

	S	M	L	XL
	properties of the phone lattices			
density	3.20	4.21	5.33	7.41
event rate	4.46	7.93	14.6	31.1
fan-out	3.36	3.95	4.56	5.48
lattice error rate	1.37	0.98	0.72	0.44
	FLaVoR word error rates			
baseline + filler	7.80	6.85	5.87	4.98
single error + rules	4.33	3.68	3.39	3.14

Table 2: Phone lattice statistics and FLaVoR word error rates

The phone decoder was used to generate phone lattices of four different sizes (*S*, *M*, *L*, and *XL*). The main properties of the obtained lattices are given in table 2. The density is measured as the average number of different phones (ignoring the context) in parallel per frame in the phone lattices. The event rate is the average number of arcs (context dependent phones) that start per frame. The fan-out is the average number of arcs leaving a node. The lattice error rates are the phone error rates (ins. + del. + sub.) of that path from the phone lattice that matches best with the reference transcription.

### 3.3. Layer2: Robust word decoding

As baseline experiment, we rescored the phone lattices using an FST formed by composing lexicon and LM, i.e. a transducer that maps phones to words and upholds the LM constraints. Since this baseline transducer contains no techniques to recover from ‘errors’ in the phone lattice, it fails to produce output when not a single trace in the phone lattice corresponds to a valid phone sequence in the FST. To prevent this, we added a ‘filler’ word to the lexicon consisting of a single arbitrary phone. The ‘filler’ may be observed after any word and can be followed by any word, even itself. The insertion penalty of the ‘filler’ is set high as to assure that it is only used where the baseline transducer would block. Table 1 shows the results obtained with and without the ‘filler’ when rescoring the *XL* phone lattices. All lattice rescoring was done with a pseudo time synchronous beam search decoder. Even without pruning, processing the lattices of a complete test set only took a few seconds. This shows that the low event rate of the phone lattices can significantly reduce the search effort in the subsequent decoding stages.

The assimilation rules in the morpho-phonological component of the FLaVoR-architecture provide robustness w.r.t. regular pronunciation alternations. To evaluate this aspect on RM, we created pronunciation rules to handle degemination and to (optionally) split syllabic segments into the two composing phonemes, i.e. a schwa followed by an /n/, /m/, /l/ or /r/. The probability of any alternation w.r.t. the canonical form was set (arbitrarily) to 10%. The rules were kept rudimentary since the tests on RM only serve as proof of concept<sup>2</sup>. Moreover, we expect the phone lattices to contain the non assimilated phones for more subtle pronunciation variations such as voicing assimilation. The search space now consists of the pronunciation rules composed with the lexicon (including the ‘filler’ word) and the LM. As shown in table 1, modeling the pronunciation variation has a positive effect, but is by itself insufficient to lower the WER to that of an all-in-one decoder.

<sup>2</sup>Ultimately, the goal of the two layer approach is the incorporation of more precise linguistic models that are, to a certain aspect, generic for a language. In the FLaVoR project, both accurate morpho-phonological and morpho-syntactic models are being developed for the Dutch language.

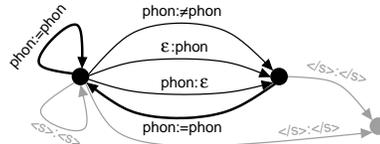


Figure 2: ‘Single error’ FST – the ‘error’ arcs use thin lines, the gray items handle sentence start and ending

As first technique to cope with non-regular alternations, we allowed arbitrary phone insertions and deletions, i.e. we allow (1) any phone in the lexicon (after applying the assimilation rules) to be deleted and (2) an arbitrary number of insertions before and after any phone in the lexicon. Insertions and deletions are controlled with a single parameter each, i.e. the costs are phone independent. Substitutions are not modeled explicitly since they can be seen as an insertion followed by a deletion (note: the cost of a substitution is not a free parameter), and because we assume that likely substitutions are already present in the phone lattice. As this technique can cope with arbitrary alternations, the ‘filler’ word is no longer needed. Table 1 indicates that this simple scheme is reasonably effective but still not able to completely bridge the gap w.r.t. an all-in-one approach. A major down side of this simple scheme is that it leads to slow decoding: by following a sequence of phone deletions, likely search tokens spread to many new locations in the search space. In an all-in-one decoder arbitrary deletions are not allowed, and hence this spreading a likely tokens is effectively counteracted by the minimal duration constraints imposed by the 3 state left-to-right phone HMMs.

In order to further lower the WER, we need more accurate modeling of the error phenomena. Yet, at the same time we need to impose extra constraints on the error model to prevent excessive spreading of the likely tokens. We satisfy both requirements with the ‘single error model’. The more accurate modeling is obtained by allowing insertions, deletions and substitutions, and by having a complete cost matrix (i.e. phone dependent costs). The extra constraint comes from the fact that after any ‘error’, we require the next phone to be correct. In other words, an arbitrary substitution, insertion or deletion can occur anywhere in the phone lattice as long as it is isolated by at least one correct phone before and after the location of the alternation. Figure 2 shows the FST with which the baseline transducer must be composed to allow ‘single errors’. The probabilities in the phone alternation matrices (ins. + del. + sub.) were estimated using a Viterbi style training that maximized the likelihood of phone lattices generated on the train set given the automatically generated reference phone transcription. During this procedure, we noticed a strong difference between the phone lattices produced on the train set and those produced on the test sets: the lattice error rate for the train set is, depending on the size of the lattice, a factor 5 to 10 lower than that of the test sets. This is a direct consequence of the maximum likelihood (ML) training of the acoustic models: the likelihood of data point in the train set must be made as high as possible, and since probability densities must integrate to 1.0, the likelihood of all other data points (including those in the test sets) must be as low as possible. Consequently, data points from the train set tend to favor the HMM states they were assigned to during the training, which in turn strongly biases the output of the phone recognition towards to the phone transcription used for the training. Given the different behaviour between test and train data, we enforced a high degree of smoothing on the probabilities in the phone alternation matrices.

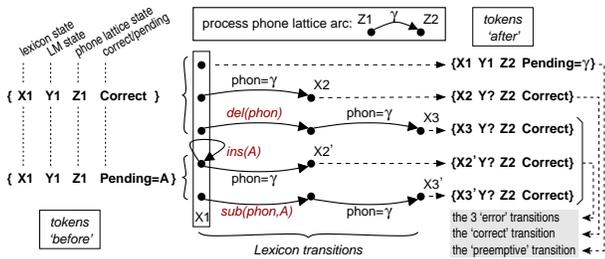


Figure 3: Efficient implementation of the ‘single error model’

### 3.4. Discussion

Table 1 shows that despite the crudeness of the resources used (over simplistic assimilation rules, ill estimated probabilities in the phone alternation matrices and various other sub-optimality), the FLVoR-decoder is capable of achieving results as good as those of an all-in-one decoder. Right now, the crudeness of the resources necessitates the use of large phone lattices. However, as can be seen in table 2, the addition of better (more realistic) schemes to achieve robust word decoding reduces the impact of lattice density on the final WER.

In order for the FLVoR-approach to reach its full potential, several sub-optimality have to be remedied. First, we need better estimates for all probabilities in the assimilation rules and the error model. This requires an accurate phonetic transcription of the train database. The automatic methods we use right now are hampered by the strong bias the acoustic models have when handling training samples. Leaving-one-out techniques should solve this. Since in ML training each speaker corresponds to a set of occurrence counts, it may suffice to remove the counts of the train speaker at hand, hence eliminating the need for full leaving-one-out training.

A second improvement lies in the acoustic models. The current models were designed for an all-in-one decoder, i.e. all cross-word effects are simply trained into the context-dependent phone models. To be optimal for a FLVoR-approach, at least the regular pronunciation alternations should be removed from the acoustic models by using assimilation rules during the training. It should even be possible to use a FLVoR-style forced alignment (using the ‘error’ model) on the train set to resolve the non-regular phenomena, hence obtaining ‘uncontaminated’ phone models.

A last sub-optimality lies in the configuration of the phone recognizer. Tuning this component to achieve minimal error rates may not be optimal since this favors phone deletions. Tuning for fewer deletions may reduce the reliance on the error model to cope with the deletions, and hence improve the WER.

## 4. Efficient implementation

While the prototype uses FST techniques throughout, the final system will rely on FST’s for the morpho-phonological component (lexicon and pronunciation rules) only. Limiting the morpho-syntactic components to FST’s would severely limit the type of modeling that could be used. In fact, even our current all-in-one decoder allows almost any type of LM by means of a dynamic integration of the LM info [8].

It is feasible to incorporate the ‘single error model’ in the search by means of an FST – we used this technique for all experiments reported on in this paper. However, this is very inefficient since the knowledge that every ‘error’ must be followed by

an exact match cannot be exploited. A direct implementation of the error model as depicted in figure 3 is far more efficient. The main ‘trick’ is to delay the traversing of the insertion and substitution arcs in the error model until a next phone lattice arc is processed. Hence, any error transition is processed in conjunction with the required subsequent match. The explicit requirement of a subsequent match assures that only few insertion and substitution transitions will be possible, thus reducing the overhead of the error model drastically. In order to delay a transition, an extra ‘pending’ state has to be introduced. Hence, tokens in the search are now described by (1) the lexicon state, (2) the LM state, (3) the state in the phone lattice, and (4) a pending/correct state where a pending state also contains the phone that was withheld. Propagating a token requires testing and/or taking few possible transitions: three error transitions, the ‘correct’ transition, and a ‘preemptive’ transition needed to delay the error transitions.

## 5. Conclusions

We presented a flexible, layered architecture for LVCSR which isolates the low-level acoustic-phonemic search from the the higher layers by means of a dense phone network. In this paper, we focused on the second layer: the word decoder. We proposed a robust and an efficient decoding strategy that achieves results as good as those of an all-in-one decoder, and yet at the same time still has substantial room for further improvements.

## 6. Acknowledgements

The research reported in this paper was funded by IWT in the GBOU program, project FLVoR (Project number 020192) – <http://www.esat.kuleuven.ac.be/psi/spraak/projects/FLVoR>.

## 7. References

- [1] J.J. Odell, *The Use of Context in Large Vocabulary Speech Recognition*, Ph.D. thesis, University of Cambridge, U.K., March 1995.
- [2] M. Mohri, “Finite-state transducers in language and speech processing,” *Comp. Ling.*, vol. 23, no. 2, pp. 269–311, 1997.
- [3] K. Demuynck, J. Duchateau, and D. Van Compernelle, “A static lexicon network representation for cross-word context dependent phones,” in *Proc. EUROSPEECH*, Rhodes, Greece, Sept. 1997, vol. I, pp. 143–146.
- [4] C.L. Wayne, “BAA #02-06: Effective, Affordable, Reusable Speech-to-text (EARS - DARPA/ITO),” 2002.
- [5] K. Demuynck, T. Laureys, D. Van Compernelle, and H. Van hamme, “Flavor: a flexible architecture for LVCSR,” in *Proc. EUROSPEECH*, Geneva, Switzerland, Sept. 2003, pp. 1973–1976.
- [6] H. Ney and X. Aubert, “Dynamic programming search strategies: From digit strings to large vocabulary word graphs,” in *Automatic Speech and Speaker Recognition*, C. Lee, F.K. Soong, and K.K. Paliwal, Eds., pp. 385–411. Kluwer Academic Publishers, 1996.
- [7] O. Scharenborg, D. Norris, L. ten Bosch, and J.M. McQueen, “How should a speech recognizer work?,” *Cognitive Science*, vol. 29, no. 6, pp. 867–918, 2005.
- [8] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*, Ph.D. thesis, K.U.Leuven, ESAT, Feb. 2001.