A LAYERED APPROACH FOR DUTCH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Joris Pelemans, Kris Demuynck and Patrick Wambacq

Katholieke Universiteit Leuven - Dept. ESAT
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{joris.pelemans,kris.demuynck,patrick.wambacq}@esat.kuleuven.be

ABSTRACT

In this paper we investigate whether a layered architecture that has already proven its value for small tasks, works for a system with large lexica (400k words) and language models (5-grams) as well. The architecture was designed to decouple phone and word recognition which allows for the integration of more complex linguistic components, especially at the sub-word level. It was tested on the Dutch language which - with its large variety of accents and rich morphology - is ideally suited to benefit from this integration. The results reveal that the architecture is already competitive to an all-in-one approach in which acoustic models, language models and lexicon are all applied simultaneously. Candidates for further improvement to the system based on a conditional phone confusion model are suggested.

Index Terms— LVCSR, phone lattice decoding, ASR architecture, phone confusion matrix, accented speech

1. MOTIVATION AND AIMS

The current mainstream approach to automatic speech recognition is to combine all knowledge sources, acoustic models (AMs), language models (LMs) and lexicon, into one huge search space. This all-in-one approach has considerable advantages. First, it has been well developed and proven to be a reliable method for all kinds of tasks. Second, by immediately integrating the lexicon and LM, it is able to prune the large amount of confusion in the acoustic signal based on all knowledge sources. However, the monolithic search strategy has some disadvantages as well. Integrating all knowledge sources at once makes for a complex task which means that the knowledge sources all have to be kept simple. Consequently, almost all recognizers employ non-optimal linguistic components such as static lexica (lexicalization of morphological processes) and N-gram LM's. Furthermore, since the AM operates from left to right, it enforces this mode of operation on the other models as well. This is often inefficient for decisions which (partially) depend on a right context, for example the LM. Third, only the knowledge sources that fit the Hidden Markov Model (HMM) paradigm can be readily included. Duration, prosody and cross-frame properties in general are much more difficult to exploit. Finally, when targeting Large Vocabulary Continuous Speech Recognition (LVCSR), lexica grow and LM perplexities increase and hence the impact of integrating the lexicon and LM at an early stage diminishes.

These factors were the motivation to develop a new architecture called FLaVoR (Flexible Large Vocabulary Recognition) [1] in which the decoding is split into two layers. A first layer takes care of the acoustic recognition to output a dense phone network. The output of this layer serves as input to a second layer in which the lexicon and LM are used to do word decoding. Decoupling the two layers makes it possible to incorporate cross-frame information after phone recognition and to integrate more complex models in the word recognition stage. This approach has been proven to match the standard all-in-one approach for the English Wall Street Journal test suite [1, 2, 3]. This task however was limited to read, noise-free speech and was performed using relatively small lexica (<20k words) and LMs (bigrams).

The main aim of this research is to show that a basic FLaVoR setup can compete with an all-in-one approach for systems with large lexica (400k words) and LMs (5-grams), and that it can handle speech that is spontaneous and noisy. The setup can then be used to exploit FLaVoR's flexibility to further improve its accuracy.

FLaVoR was especially designed to contrast current architectures by introducing more advanced linguistic knowledge at the sub-word (syllables, morphemes) level which makes most sense for generative languages like Dutch or German or for strongly agglutinative languages like Turkish and Finnish, as opposed to English where the words are more or less atomic entities and hence operating at the word level is practical. Therefore, the second aim of this research is to test the system on a language that would benefit more from the FLaVoR design than English. Dutch is a morphologically productive language which uses inflection, derivation and compounding to produce new words. It also has a high rate of foreign words and a large variety of different accents. These properties make it well suited for the FLaVoR architecture.

The paper is organized as follows. Section 2 describes the FLaVoR architecture in more detail. In section 3 we introduce the task, explain the corresponding setup of the FLaVoR system and talk about the experiments we did for both the phone and word decoding layers. We end with a thorough discussion of our results in section 4.

2. THE FLAVOR ARCHITECTURE

In this section we briefly recapitulate the FLaVoR architecture, shown in Figure 1. For more details, the rationale behind FLaVoR and differences with existing multi-pass strategies, we refer to [1].

2.1. Layer 1: Phone decoding

In the first layer of the FLaVoR system a phone decoder determines the network of most probable phones given the acoustic features of the incoming signal. The knowledge sources employed are an AM and a phone transition model, i.e. a LM for phones. The resulting

This research was supported by the CLARIN projects TTNWW and SPRAAK2TAAL.

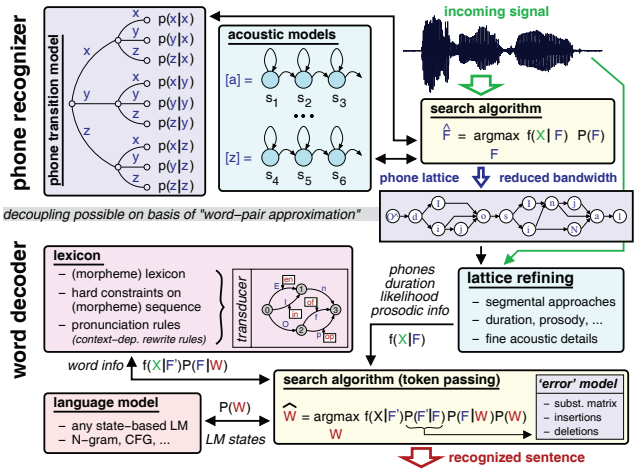

Fig. 1. The FLaVoR architecture

phone network can be enriched with meta-data such as prosody or speaker identities in order to provide rich information to the second layer. The meta-data is not restricted to the HMM paradigm which opens up opportunities to incorporate cross-frame information.

2.2. Layer 2: Word decoding

The goal of the second layer of our FLaVoR system is to adopt the phone lattice as input and map phone sequences onto words without enforcing the left-to-right operation that is typical for acoustic decoding. As a prototype, phone-to-word mapping is achieved by transforming the lexicon and LM into a Finite State Transducer (FST) and applying this to the phone lattice. It has been shown that such transducers are a very compact and efficient solution for decoding [4].

The decoupling of acoustic and word decoding not only relaxes the constraints on the mode of operation of the linguistic models, but it also reduces the number of parallel options each input stands for. While a monolithic search engine has to match all incoming feature vectors with all possible combinations of phones and end positions at that point in the search, the phone network will only contain the set of best matching phones with their optimal start and end times.

Since the FST cannot recover from the early acoustic pruning in the first layer, error correction has to be applied. An error model is built to find the typical phone confusions in a language, creating a confusion matrix with costs for each phone substitution, insertion and deletion. This matrix is created by retraining an initial confusion matrix based on phonetic properties on a huge corpus using EM optimization. The error model could be incorporated in the search by means of an FST, but as was explained in [2], a dedicated implementation was chosen for efficiency reasons.

3. EXPERIMENTS

3.1. Task and reference results

For the purpose of testing Dutch spontaneous speech the N-Best evaluation benchmark [5] was chosen. N-Best contains Northern and Southern Dutch, broadband and telephone speech, totaling to 260 hours of training data. For each of the four subtasks, the amount of speech consists of 1 to 2 hours of development data and 2 to 3 hours of evaluation data. The evaluation data is known to be different from both the training and development data in a sense that it contains considerably less telephone speech and more accented and spontaneous speech. For most systems this results in a big difference in WERs as audio normalization and adaptation were not the focus of development [5, 6, 7]. As a reference we used the results of our all-in-one system [7] developed for Southern Dutch. Because the AMs for telephone speech are very different from those for broadband speech we decided to limit the task to the broadband speech for the time being. This reduced the amount of speech to 40h of training data, 55 minutes of development data and 1h55 of evaluation data. If the architecture proves to be fruitful, experiments with telephone speech and Northern Dutch can be performed at a later stage.

3.2. Setup

In this section we briefly discuss the models we employed in our system. For more information we refer to [7].

3.2.1. Acoustic modeling

For all experiments we used our in-house state-of-the-art speech recognition system [8]. For the AMs, 49 three-state acoustic units (46 phones, silence, garbage and speaker noise) and one single-state phone (short schwa) are modeled using our default tied gaussian approach, i.e. the density function for each of the 4k cross-word context-dependent tied states is modeled as a mixture of an arbitrary subset of gaussians drawn from a global pool of 50k gaussians. The mixtures use on average 180 gaussians to model a 36 dimensional observation vector of MIDA features [8]. These were obtained by means of a mutual information based discriminant linear transform (MIDA) on vocal-tract length normalized (VTLN) and mean-normalized MEL-scale spectral features and their first and second order time derivatives (the lowest and highest MEL filter bank outputs are removed). The models were trained for an all-in-one system, i.e. the context dependent phone models are trained to cope with most of the pronunciation variation.

3.2.2. Language modeling and Lexicon

Using a lexicon of 400k words, 5-gram word LMs with modified Kneser-Ney discounting were trained on 4 main text components: 12 Southern Dutch newspapers, 10 Northern Dutch newspapers and transcriptions of broadcast news and conversational telephone speech. The text components were interpolated linearly and perplexity minimization was done to find the optimal interpolation weights. Lexicon creation was handled by an updated version of the system described in [9]. Dutch has a decent amount of (regional) pronunciation variation. This issue was addressed by using phonological rules to generate the likely pronunciation variants. This resulted in a median of 3.8 pronunciations per word or 1.13 variants per phone in the canonical word transcriptions.

3.2.3. Post-processing

Since Dutch compounds are always written as 1 word, the word recognition results were post-processed for compounding. Two subsequent words were replaced by their compound if the following criteria are met: 1) the words are longer than 3 letters, 2) the words are not very rare, 3) the unigram count of the compound is higher than the bigram count of the individual words. This approach essentially extends the 400k lexicon to a 6M lexicon.

3.3. Experiments

This section outlines the different experiments we ran for both the phone and word decoding layer. All parameters were optimized on the development data only. All timings were obtained using an Intel Core i5-2400 processor with 1 core only.

3.3.1. Phone decoding

To optimize a phone decoder, a reference phonetic transcription of the data is needed to test the accuracy of the system. Most databases, including N-Best, however only have an orthographic transcription, so it's necessary to convert this orthographic transcription into a phonetic one. As was shown in [9] a reliable way of doing this automatically is to create a pronunciation network by looking up the pronunciation for each word in the lexicon. Some additional language-specific pronunciation rules were applied to account for cross-word phenomena and typical word internal assimilation processes. The Viterbi algorithm was used to find the best path through the resulting network.

Applying the same technique on the 40h of training data, a phone transition model was estimated. We created a 4-gram and tested 3 different discounting methods: Witten-Bell (WB), Good-Turing (GT) and (modified) Kneser-Ney (KN).

Our FLaVoR system has 4 parameters to be optimized during the phone recognition layer. To combine the scores of the AM and the phone transition model we employed our standard way of handling this problem [8], by having a LM scaling factor and a word startup cost. Beam search pruning was applied to control the amount of hypotheses in the network [10]: a threshold indicates how much the score of a hypothesis can drop below the score of the most likely hypothesis; if most hypotheses have a similar score, a beam width parameter is then set to indicate how many hypotheses can be retained, keeping only the best ones.

With optimal parameters the GT discounting achieved a phone error rate (PER) of 14.70% and was used in the following experiments, although the differences with the WB and KN discounting were dismissable. Since it is uncertain that the best phone sequence hypothesis yields an optimal word sequence, we investigated our lattice with respect to the amount and quality of phone hypotheses it contains. Additional statistics were calculated and can be found in Table 1. The density of the lattice is measured as the average number of different phones (ignoring the context) in parallel per frame in the phone lattice. By relaxing the pruning parameters, lattices of different densities were made (Small, Medium, Large, eXtraLarge) to find which phone lattice density yields the best result at a reasonable speed in the second layer. To get an early indication of the quality of all hypotheses in each lattice we also calculated their lattice error rate. The lattice error rate is the PER of the path that aligns best with the reference transcription. Finally the processing time for each of the lattices was included as the real time factor (xRT).

3.3.2. Word decoding

As a baseline experiment, we created an FST consisting only of lexicon and LM, applied it to the different lattices and optimized the different parameters of our word decoder for each of them. The parameters are the same as in the first layer of our system: score combination and beam search pruning parameters. When applying the transducer to the lattice, it becomes clear that not every sentence reaches a valid end state: if no valid word sequence can be found in the phone sequence network, there is no way for the FST to recover from this error. In order to overcome this problem and to further improve results, some form of error correction is necessary to provide the possibility of substituting, inserting or deleting phones. In fact, the role of the error model is to bridge the gap between the expected pronunciation as stored in the lexicon and the observed pronunciation as recorded in the phone lattice. In practice we train an error model based on a large corpus to find typical confusions, i.e. typical mistakes the recognizer (and often also humans) makes, by comparing the output of the recognizer with the transcription. Confusions that are very common e.g. substituting the two fricatives 's' and 'f' will be given a low cost while very unlikely confusions e.g. substituting the vowel 'a' and the consonant 'd' will be very costly. For every possible insertion, deletion and substitution we calculate its cost to end up with a full confusion matrix containing all the costs. A "single error" constraint is set to prevent the huge growth of hypotheses: after each error the next phone is required to be correct. By allowing only a single error in a row the recognized word sequence cannot deviate too much from the phone sequence hypotheses in the lattice.

Training the error model was done by making an initial confusion matrix based on the phonetic properties of every Dutch phone. To optimize the cost of every substitution, insertion and deletion we created a phone lattice of the N-Best training corpus using the first layer of our system and then used this lattice to estimate the optimal costs by Expectation Maximization (EM), given the initial matrix and the reference transcription.

	S	M	L	XL
density	4.37	5.41	6.52	7.64
lattice error rate	1.53	1.24	1.09	1.00
processing time (xRT)	0.25	0.31	0.37	0.40

Table 1. Phone lattice statistics for the development data

		dev		eval	
		xRT	WER	xRT	WER
all-in-one		0.50	6.29	0.88	19.94
		1.38	5.71	2.57	19.16
		3.56	5.45	9.49	18.80
FLaVoR without error model	S	0.41	11.08	0.68	23.61
	M	0.55	9.61	0.86	22.49
	L	0.70	8.88	1.03	21.85
	XL	0.84	8.16	1.41	21.54
FLaVoR with error model	S	3.42	5.81	4.27	19.19
	M	3.70	5.81	4.55	19.15
	L	3.93	5.79	5.13	19.01
	XL	4.17	5.63	5.69	18.96
FLaVoR with pruned error model		1.67	5.76	2.72	19.26

Table 2. WERs and processing times for all-in-one and FLaVoR

4. DISCUSSION AND CONCLUSION

During our experiments we found that the FLaVoR approach is very robust with regards to the phone recognition layer. Changing the task, the phone discounting method or the preprocessing hardly has any effect on the optimal parameters of the first layer. Furthermore, as can be seen in Table 1, the creation of the phone lattices, which is a fixed cost, is faster than in real time, even for the XL phone lattice. This means that making changes to the phone layer is very easy and fast. Moreover, the generic nature of the first layer allows

it to 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 [3]), for handling specific problems (e.g. recognition of proper names) or for keyword spotting.

As depicted in Table 2 the FLaVoR approach without the error model is very fast for both development and evaluation data, with processing times not much higher than the creation of the phone lattices. The obtained WERs with this setup are still acceptable considering the fast decoding, but incorporating some form of error modeling is preferable. The WERs clearly improve when increasing the density of the phone lattice, at the cost of an increase in processing time.

When error models are employed, the differences in WERs and processing times between the various lattices are less pronounced, especially in the development data. We believe that the current error model is powerful enough to cope with a mild amount of confusability in the data, even when only a limited number of original hypotheses are included in the phone lattice. Since the evaluation data contains more spontaneous and accented speech, its confusability is considerably higher and the word decoding layer can still gain from extra phone hypotheses. Better tuned error models should be able to handle the increased confusability, thus eliminating the need of higher densities and improving the processing time as well as the WER.

When comparing our WERs with the all-in-one approach we see that for each lattice the WER is competitive for both development and evaluation data. Moreover, with regards to decoding speed it should be noted that our system handles the discrepancies between development and evaluation data much better. When using an all-in-one approach the processing times almost double or even triple, while ours differ only by a factor 1.2 to 1.4. WER optimization was done for every lattice without taking decoding speed into account. For better time comparison, the error model was pruned by ignoring phone operations that have too high a cost compared to a threshold value. This pruned model yields the best combined results of WER and decoding speed on the XL and L lattices for development and evaluation data respectively. Again the results are competitive, but given access to all the knowledge sources the all-in-one approach currently still has an advantage. Analysis of both decoders showed however that when it comes to investigating the different explanations according to the LM, the FLaVoR approach definitely wins, i.e. it will benefit more from more powerful LMs. Moreover, there are a lot of unexplored opportunities to further improve our system.

In all our experiments we limited ourselves to 4-gram phone transition models. It is likely that enlarging the phone context has a positive impact on both phone and word recognition results. However care must be taken to avoid overfitting since there is only a limited amount of training data.

Another standing issue is how pronunciation variations can best be handled. In the current setup, this task is divided rather arbitrary between the AM (the phone models were trained based on canonical lexicon pronunciations), the lexicon (pronunciation rules) and the error model. Optimizing the role of each component (which component models which part of the variation) has the potential to both improve the recognition and to speed up the decoding.

It is clear that the error model has a big impact on the result, both in WER and processing time. Improving this model will not only lead to lower WERs, but also to less hypotheses to consider, thus making the whole system a lot faster. The ideal model we want to approach will have only a little overhead compared to the system without an error model. This will provide the opportunity to incorporate even more complex knowledge sources in the second layer.

Likely candidates for improving the error model consist of conditional substitutions, insertions and deletions. One possibility would be to take phone duration into account when considering the possible phone operations e.g. the substitution of a long vowel into another one should be more costly than the substitution of a short vowel and its deletion should only be allowed at a very high cost. Optimal duration boundaries should be investigated for all phones.

Phone context is a second possible upgrade of which we believe the error model and hence the WERs and times will benefit. In spontaneous speech, human pronunciation is typically very sloppy which results a.o. in vowels, even long ones, being substituted by the neutral schwa. Our error model correctly estimates this behavior, consequently assigning low costs to these substitutions. Our phone transition model however indicates that in Dutch the schwa can appear after a diphthong with a relatively high probability, while other vowels, especially long ones, are very rare if not impossible to appear in this context. We believe similar phenomena exist for larger contexts, as well as right contexts.

As a main conclusion we can state that the FLaVoR approach works on large systems: the results using 5-gram LMs and 400k lexica on spontaneous, noisy speech are already competitive to those of an all-in-one approach, which was the main aim of our research. The large variety of different accents and the morphological challenges in Dutch do not hamper its functionality in any way. In future work, when handling acoustic mismatches is better divided between AM, lexicon and error model, FLaVoR should have an advantage due to the ease with which the error model can be made context and speaker dependent, but even in its current form, the setup is ready to take advantage of the flexibility of the FLaVoR design.

5. REFERENCES

[1] K. Demuynck, T. Laureys, D. Van Compernolle, and H. Van hamme, "FLaVoR: a flexible architecture for LVCSR," in *Proc. EUROSPEECH*, 2003, pp. 1973–1976.

[2] K. Demuynck, D. Van Compernolle, and H. Van hamme, "Robust phone lattice decoding," in *Proc. ICSLP*, 2006, pp. 1622–1625.

[3] J. Duchateau, K. Demuynck, and H. Van hamme, "Evaluation of phone lattice based speech decoding," in *Proc. INTERSPEECH*, 2009, pp. 1179–1182.

[4] M. Mohri, "Finite-state transducers in language and speech processing," *Comp. Ling.*, vol. 23, no. 2, pp. 269–311, 1997.

[5] J. Kessens and D. A. van Leeuwen, "N-Best: the Northern- and Southern-Dutch benchmark evaluation of speech recognition technology," in *Proc. INTERSPEECH*, 2007, pp. 1354–1357.

[6] J. Despres, P. Fousek, J.-L. Gauvain, S. Gay, Y. Josse, L. Lamel, and A. Messaoudi, "The joint LIMSI and Vecsys research systems for NBEST 2008," *Language*, 2008.

[7] K. Demuynck, A. Puurula, D. Van Compernolle, and P. Wambacq, "The ESAT 2008 system for N-Best Dutch speech recognition benchmark," in *Proc. ASRU*, 2009, pp. 339–343.

[8] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*, Ph.D. thesis, K.U.Leuven ESAT, 2001.

[9] K. Demuynck, T. Laureys, and S. Gillis, "Automatic generation of phonetic transcriptions for large speech corpora," in *Proc. ICSLP*, 2002, vol. I, pp. 333–336.

[10] V. Steinbiss, B.-H. Tran, and H. Ney, "Improvements in beam search," in *Proc. ICSLP*, 1994, pp. 2143–2146.