A Bottom-Up Stepwise Knowledge-Integration Approach to Large Vocabulary Continuous Speech Recognition Using Weighted Finite State Machines

Sabato Marco Siniscalchi1, Torbjørn Svendsen2, and Chin-Hui Lee3

1Department of Telematics, Kore University of Enna, Enna, Italy
2Department of Electronics and Telecommunications, NTNU, Trondheim, Norway
3School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

{marco77, torbjorn}@iet.ntnu.no, chl@ece.gatech.edu

ABSTRACT
A bottom-up, stepwise, knowledge integration framework is proposed to realize detection-based, large vocabulary continuous speech recognition (LVCSR) with a weighted finite state machine (WFSM). The WFSM framework offers a flexible architecture for different types of knowledge network compositions, each of them can be built and optimized independently. Speech attribute detectors are used as an intermediate block to obtain phoneme posterior probabilities over which a phoneme recognition network is designed. Lexical access and syntax knowledge integration over this phoneme network are then performed to deliver the decoded sentences. Experimental evidence illustrates that the proposed system outperforms several hybrid HMM/ANN systems with different configurations on the Wall Street Journal task while it is competitive with conventional LVCSR technology.

Index Terms— Weighted finite state machine, speech attribute detection, speech knowledge integration, ASR

1. INTRODUCTION
Hidden Markov models (HMMs) have become the predominant modeling approach to address the automatic speech recognition (ASR) problem. Good ASR accuracies are often achieved by using a large amount of transcribed task-specific speech data and text material to train high performance acoustic and language models for building large-scale, integrated finite-state network representation of the recognition tasks. Nonetheless, HMM-based ASR system seem to be close to reach their limit of performance, and alternative approaches have been proposed in recent years (see [1] for a brief survey).

ASAT [1] and SIRKUS1 approach ASR from a bottom-up, acoustic-phonetic perspective by developing a detection-based framework based on speech attribute detection and knowledge integration. It was demonstrated that high-accuracy phoneme recognition can be achieved within this framework [2]. Moreover, a lattice rescoring approach for integrating acoustic-phonetic information obtained with a bank of speech attribute detectors into ASR was also crafted in [3] with good results. Nevertheless, the ultimate goal of ASAT/SIRKUS is to build stand-alone, bottom-up, detector-based LVCSR algorithms that can better use the information gathered in the detection stages in order to deal with situations in which not all knowledge sources required for ASR can be pre-compiled and integrated in finite state network representation of the recognition task.

In this paper, we present our first attempt at designing a stand-alone LVCSR system by integrating the detection-based approach into the WFSM framework [4]. In a step-by-step procedure LVCSR is carried out in a bottom-up fashion by performing lexical access and syntax knowledge integration over the output of our detection-based frontend, which generates frame-level speech attribute detection scores and phoneme posterior probabilities. The final system has several stages, and additional information can be integrated at each stage to obtain a refined set of hypotheses, like initially envisioned in the ASAT paradigm [1]. The proposed technique is evaluated on the WSJ0 corpus [5], and the experimental results demonstrate the feasibility of this approach. Our decoupled LVCSR system achieves word error rates (WERs) that is competitive with standard hidden Markov model (HMM) based LVCSR algorithms. Furthermore, it outperforms standard HMM/ANN systems, and thus represents an interesting alternative solution to the use of highly-accurate phoneme posterior probabilities for LVCSR. Finally, it is interesting to point out that the accurate phoneme posterior estimates licenses certain forms of early pruning, e.g., [6]), that allows us to overcome the large memory requirement when the whole input utterance is represented as a WFSM.

The paper is organized as follows. First, the overall system is presented. Section 3 describes the experimental results, and Section 4 concludes the study.

---

1This work is part of the SIRKUS project.
Fig. 1. The proposed bottom-up stepwise LVCSR system.

Fig. 2. The detection-based frontend. Each attribute detector analyzes any given input frame and produce a posterior probability score. The Append & Expand module stacks together attribute posteriors to take into account long-term dependencies. The merger delivers phone posteriors.

2. OVERALL SYSTEM

The proposed LVCSR system consists of three main modules: (1) a detection-based frontend that scores speech frames, (2) a module that combines these scores and generates a lattice at a word level, and (3) a language model (LM) rescoring block. The overall system is a bottom-up stepwise word decoder (see Figure 1), and these blocks will be explained in more detail in the following sections.

2.1. Detection-based Frontend

The detection-based frontend consists of two main blocks: a bank of speech attribute detectors, and an evidence merger (see Figure 2). An attribute detector is built for each of the following 21 phonetic features: fricative, approximant, nasal, stop, vowel, coronal, dental, glottal, high, labial, low, mid, retroflex, velar, anterior, back, continuant, round, tense, voiced, and silence. Each attribute detector analyzes a frame of the input speech signal and produces the posterior probability that pertains to some acoustic-phonetic attribute. Feed-forward multi-layer perceptrons (MLPs) with one hidden layer and 800 hidden nodes are used to build detectors. Detectors are trained on short-time spectral features and learn the mapping from the acoustic to the attribute space. The number of inputs to each detector is 39 (see Section 3.2). The number of outputs is two: attribute present, and attribute absent. Long-term dependencies among attributes are taken into account in the Append & Expand module, which stacks together a window of eleven frames around the frame to be classified and generates a supervector. This approach was shown effective in [7]. The merger is then fed with this supervector. The merger is implemented with a frame-based MLP with a single hidden layer having 1500 nodes. The MLP discriminates among 40 phoneme classes, as discussed in the experimental section.

For all MLPs, the sigmoid and softmax non-linearities are used as activation functions in the hidden and output nodes, respectively. The standard back propagation algorithm is adopted as training method, and to avoid over-fitting, the reduction in classification error on a development set during the training phase is chosen as a stopping criterion.

2.2. WFSD & LM Rescoring Modules

In this work, the WFSD approach is adopted to deliver word-level evidence directly from the frame-level scores generated by the detection-based frontend, in a bottom-up fashion. To this end, the output of the detection-based frontend, for a given sentence, needs first to be represented as a feature acceptor (F). Figure 3 (a) displays a simplified feature acceptor for a speech sentence with six frames. F is practically a sausage with a number of states that equals the length of the input sentence (in frames), and a number of edges between each pair of states that equals the output dimension of the merger (i.e., the number of phonemes to be classified). The weight carried by each transition edge is equal to \(-\log_e(Prob(Phoneme | SpeechInput))\), so it is a positive
network. To avoid cluttered figures, only a few edges are displayed. Word recognition can now theoretically be performed by lexical access to a vocabulary of words along with integration of syntax knowledge. These two tasks are accomplished by composing the F acceptor with a lexicon transducer (L) and a back-off language acceptor (G). For the sake of completeness, it must be pointed out that F is first combined with a duration transducer (D) that forces the minimum phone duration to three frames. Practically, D fuses frames to generate a single phone unit as shown in Figure 3 (b.) More detail on L and G can be found in [4].

A WFSM that maps from frame distributions to word strings restricted by G can be obtained through the composition operation [4]: $F \circ D \circ L \circ G$, where $\circ$ indicates the WFSM composition operation. It is thus utilized to combine all levels of knowledge integration in our ASR system into an integrated recognition network (RN) in a convenient, efficient and general manner. It should be recalled that WFSM is an elegant framework to integrate several knowledge sources in a pre-compiled RN, yet it has the side effect of requiring a very large amount of memory at run time when decoding is performed by representing the input utterance itself as a WFSM. This is the case scenario within which we are operating, since the utterance is represented as the sausage F. Viterbi-based decoders, such as the AT&T drecog, or the Juicer decoder [8] cannot thereby be employed, so the best-path search needs much more memory than in common practice. Therefore, some ad-hoc decoding strategies need to be devised to accomplish the word recognition task. The highly-accurate phoneme scores, see Table 2, allow us to devise a decoupled knowledge integration scheme: $((F \circ D) \circ L) \circ \text{bigram-G}$ followed up by trigram LM rescoring. A pruning process at the output of each composition step is performed to remove highly unlikely search paths and thus reduce the size of the intermediate RNs. In the first stage, $F \circ D$ is accomplished and a phoneme-level RN is generated, pruned, and then sent to the L transducer. The next combination is therefore $(F \circ D) \circ L$ that produces a word-level RN by integrating lexical knowledge. After pruning, this word-level network is composed with bigram-G to integrate LM information. A grammar-constrained, word-level RN is thus generated. Figure 4 shows the decoupled LVCSR system with WFSMs. The output WFSM module, namely a pruned word-level RN, is sent to the LM rescoring module which uses a more detailed trigram back-off LM to improve the accuracy of the overall system. The output of this step is a word-level lattice over which the best path is computed and delivered.

### 3. EXPERIMENTS

#### 3.1. Experimental Setup

All experiments were conducted on the 5,000-word speaker independent WSJ0 (5k-WSJ0) task [5]. The parameters of the detectors were estimated using training material from the SI-84 set (7077 utterances from 84 speakers, i.e., 15.3 hours of speech). A cross-validation (CV) set was generated by extracting 200 sentences out of the SI-84 training set. The CV set accounts for about 3% of the SI-84 set and was used to terminate the training. The remaining 6877 SI-84 sentences were used as training material. Evaluation was carried out on the Nov92 evaluation data (330 utterances from 8 speakers). As common practice, 13 MFCCs + deltas+ delta-deltas were chosen as the short-time spectral representation of the speech signal. For comparison, a HMM/GMM LVCSR system, we refer to this system as GMM-SYS, was built using with HTK toolkit. This system was based on tied-state cross-word triphone models and a trigram language model; 2818 shared states were obtained with a phonetic decision tree and each state observation density was modeled by a GMM with 8 mixture components. The HMM parameters were estimated using the classical MLE and the 6877 training sentences. Furthermore, an hybrid HMM/ANN is also built using the phoneme posteriors estimated by the detection-based frontend and a trigram language model. We refer to this system as ANN-SYS. Finally, the AT&T FSM toolkit [2] was used to build the WFSM modules.

### Table 1. Some attribute accuracy rates on the Nov92 data.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Frame Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fricative</td>
<td>93.17%</td>
</tr>
<tr>
<td>Nasal</td>
<td>95.88%</td>
</tr>
<tr>
<td>Vowel</td>
<td>94.02%</td>
</tr>
<tr>
<td>Dental</td>
<td>98.90%</td>
</tr>
<tr>
<td>Mid</td>
<td>90.70%</td>
</tr>
</tbody>
</table>

### Table 2. Phone accuracy rates (PARs) at a frame level.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>CV</th>
<th>Nov92</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame-based PAR</td>
<td>88.08%</td>
<td>83.90%</td>
<td>84.12%</td>
</tr>
</tbody>
</table>

---

2http://www.research.att.com/sw/tools/fsm/
Table 3. WER for several LVCSR systems on the Nov92 task. The performance of LOQ-1 and LOQ-2 as given in [9] are reported for comparison. DET-SYS is the proposed system. GMM-SYS and ANN-SYS indicate the HMM/GMM and HMM/ANN systems, respectively.

<table>
<thead>
<tr>
<th>System</th>
<th>WER</th>
<th>GMM-SYS</th>
<th>ANN-SYS</th>
<th>LOQ-1</th>
<th>LOQ-2</th>
<th>DET-SYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ0</td>
<td>5.5%</td>
<td>8.3%</td>
<td>8.4%</td>
<td>6.5%</td>
<td>6.0%</td>
<td></td>
</tr>
</tbody>
</table>

3.2. Results & Discussion

Table 3 shows the performance of all LVCSR systems implemented in this work. Moreover, to assess the quality of the ANN-SYS, the performance of two HMM/ANN systems trained on SI-84 and evaluated on the Nov92 reported in [9] are also included. These systems differ in their MLP architectures, and are denoted LOQ-1 and LOQ-2, respectively. The proposed bottom-up, step-by-step LVCSR system (DET-SYS) attains a WER of 6.0% (as shown in Table 3), a relative improvement of 27.7%, 28.5% and 7.7% over ANN-SYS, LOQ-1, and LOQ-2, respectively. This result demonstrates the feasibility of our approach, because it attains a much better performance than that of the hybrid HMM/ANN technique. Furthermore, by comparing the performance of all above mentioned system with the performance of the GMM-SYS, which delivers a WER of 5.5%, we can also conclude that our approach shows more competitive results with the standard HMM/GMM technology than the conventional HMM/ANN approaches. We believe that our approach can deliver even more promising results using alternative composition scheme and pruning strategies, which are currently under investigation.

4. CONCLUSION

In this study, a bottom-up, step-by-step, knowledge-integration approach to LVCSR using WFSMs was presented. The 5K-WSJ0 task was used to assess the performance of our proposed technique. Preliminary experimental results demonstrated that the proposed technique can effectively be adopted to accomplish LVCSR. Several insights were also revealed to better appreciate the reported results. It is interesting to note that the proposed decoupled approach naturally overcomes the memory-size problems of finite-state machines, and it also represents a state-of-the-art advancement in the use of phoneme posteriors to LVCSR.

Other compositions schemes will be investigated and a knowledge-based pruning strategy should be explored as suggested in our original bottom-up detection-based framework [1]. This proposed approach should also be tested on other tasks, especially on spontaneous speech recognition. Our preliminary results seem to indicate that this step-by-step recognition process can deliver good performance similar to that obtained with the conventional integrated decoding process in well-formed utterances. Moreover for ill-formed utterances, where partial understanding may be needed because an integrated approach is often incapable of properly representing the overall knowledge sources [10], we expect our framework to be robust and give a better performance.

5. REFERENCES


Note that the full SI-84 material was used to train the LOQ-2 and LOQ-2 systems, which means a 3% additional amount of training data.