Abstract

We present a two stage automatic speech recognition architecture suited for applications, such as spoken document retrieval, where large scale language models can be used and very low out-of-vocabulary rates need to be reached. The proposed system couples a weakly constrained phone-recognizer with a phone-to-word decoder that was originally developed for phrase-based statistical machine translation. The decoder permits to efficiently decode confusion networks in input, and to exploit large scale unpruned language models. Preliminary experiments are reported on the transcription of speeches of the Italian parliament. The use of phone confusion networks as interface between the two decoding steps permits to reduce the WER by 28%, thus making the system perform relatively close to a state-of-the-art baseline using a comparable language model.

Index Terms: Automatic speech recognition, decoding algorithm.

1. Introduction

In the recent years, there has been a revamped interest towards phone-based speech recognition. Mainly for two reasons: to reduce the complexity of the speech decoding task and to circumvent the vocabulary bottleneck of word-based automatic speech recognition (ASR).

Although state-of-the-art large-vocabulary ASR systems are able to recognize a huge number (even millions) of words, it is common practice to prune out rare words and n-grams, which constitute the largest part of the data for the language model (LM) training, in order to trade-off recognition accuracy versus computational complexity. While for many applications this is reasonable, for others, such as spoken document retrieval (SDR) [1] or keyword spotting, this is not the case. In fact, the effectiveness of document indexing is related to the informative words, which are typically unfrequent and hence, are not well represented in the LM.

To alleviate this problem, spoken document indexing based on phone strings [1], phone lattices [2], or phone confusion networks [3] have been proposed in combination with approximate string matching. Moreover, weakly constrained phone-based ASR has been also investigated as a mean to detect OOV words, by comparing its output hypotheses with those produced by a conventional word-based recognizer [4]. Finally, a detailed comparison of phone-based ASR across several languages has been carried out [5], that also evaluated the difficulty of decoding phone strings into word strings for several languages.

Phone-based ASR is a light-weight task with respect to word-based ASR, and has been pursued for long time as a mean to explore new research avenues in acoustic modeling and search techniques. In this respect, our work proposes a novel search architecture, especially suited for SDR applications, that addresses the computational complexity and vocabulary bottleneck through:

- task-independent and highly-accurate phone-recognition exploiting a 7-gram phone-based language model;
- generation of phone confusion networks from phone graphs or phone confusion matrices;
- fast word-based decoding of phone-confusion networks using large language models with full vocabulary and n-gram statistics.

The resulting architectures is outlined in Figure 1. Phone recognition and confusion network decoding are decoupled, use disjoint information, and can be performed independently. In the following sections, we detail each processing step and report experimental results comparing this approach with the FBK-irst large vocabulary speech recognizer on a transcription task of Italian parliamentary speeches.
2. Experimental Set-up

As test bed, we worked on the SITACAD corpus, containing recordings of speeches held at the Italian Parliament. Acoustic model training was carried out on the Italian Broadcast News (IBN) corpus, containing about 130 hours of manually and automatically transcribed speech [6]. Table 1 reports detailed statistics about these speech corpora. For LM training we used a 606M word corpus containing newswire and newspaper articles.

<table>
<thead>
<tr>
<th>genre</th>
<th>train</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td># hours</td>
<td>129h:30m</td>
<td>2h:52m</td>
<td>2h:43m</td>
</tr>
<tr>
<td># utterances</td>
<td>115024</td>
<td>301</td>
<td>420</td>
</tr>
<tr>
<td># words</td>
<td>1228000</td>
<td>20837</td>
<td>22219</td>
</tr>
<tr>
<td># phones</td>
<td>-</td>
<td>104145</td>
<td>111513</td>
</tr>
<tr>
<td>Avg utt. duration</td>
<td>-</td>
<td>34.3 sec.</td>
<td>23.3 sec.</td>
</tr>
</tbody>
</table>

Table 1: Statistics of speech corpora used for recognition experiments.

3. Speech-to-Phone Decoder

3.1. Acoustic front-end

The acoustic front-end adopted in the experiments reported is the same used in the TC-star system described in [7].

In particular, feature extraction embeds cepstral mean subtraction, variance normalization, and projection of acoustic features, based on Heteroscedastic Linear Discriminant Analysis. Finally, acoustic data are normalized using Constrained MLLR-based Speaker Normalization [8, 9].

3.2. Phone Decoder

Phone decoding makes use of a 7-gram phone LM, estimated on a phonetically transcribed version of our large news corpus. The LM result in a finite state network of 25M states. The phone decoder works in two steps. First, a phone lattice is built while searching for the best phone string. To augment recombination of hypotheses in the lattice, artificial transitions are inserted according to a LM of order lower than 7. Second, phone graph expansion with the original 7-gram LM is performed and a phone graph for the second decoding step is built.

The computed best phone strings provide phone error rates on the dev and test sets of 6.34% and 7.03%, respectively. Phone graphs are further used to generate confusion networks (CNs) as explained in the next section.

4. Phone Confusion Networks

We explored two different ways to compute phone confusion networks: by augmenting the best phone string with information from the phone-error confusion matrix, or directly from the phone graphs.

4.1. CN Generation from Confusion Matrix

Each phone in the best recognized string is augmented with its corresponding “most confusable” phone set, giving rise to a column of confusable phones. A corresponding phone confusion matrix is estimated on the development set from which normalized posterior scores are estimated. This approach is similar to the ones reported in [10].

This way of generating CNs does not take into account possible phone deletions occurred in the best recognized strings. Hence, the confusion network was enriched with additional optional phones by means of an hidden n-gram model [11]. Finally, CNs are pruned by removing all terms falling outside a given percentile (0.97).

4.2. CN Generation from Phone Graphs

CN can be directly generated from the phone graph as reported in [12, 13]. A straightforward application of the procedure would however result in CNs with too many alternatives in each column, which would result impossible to decode efficiently. Hence, pre-emptive pruning was applied on the phone graph as follows. Adjacent time segments were inspected and only arcs exhibiting the largest posterior probabilities in each segment were kept. In this way, given the duration information of the segments and the number of transitions to keep, it is possible to obtain pruned graphs with uniform densities along the time axis.

By generating CNs with this approach, the order of the LM constraints used to produce the phone graphs in the first decoding step becomes crucial. In fact, the application of low order LM constraints increases the recombination of phone hypotheses in the lattice, augments the average density of the resulting phone graph and, consequently, reduces the corresponding Phone Graph Error Rate (PhGER). However, phone graphs of high density require applying more aggressive pruning, to get manageable CNs from them. The application of trigram LM constraints, before phone graph expansion, provided the best trade-off between graph density and PhGER.

5. Confusion Network Decoder

Phone-to-word decoding can be addressed as a machine translation task. In this specific framework, source symbols are phones, target symbols are words, translation is monotonic and basically consists in finding the best segmentation of the phone string and for each segment the best word among possible homophones.

Phone CNs permit to represent very large sets of possible phone strings in a compact way. Each path in the CN is assigned a posterior probability, computed as the product of the posterior probabilities of the traversed phones.

In our system we employed a CN decoder [14] which has been developed for a popular phrase-based statistical machine translation toolkit, called Moses [15]. In this context we use phrases only for the input, to represent segments of phonemes, and single words on the target language. The relationship between a phone phrase and a word is given by the possibly multiple phone transcriptions of the latter, which are stored in a phrase table.

Moses computes the optimal solution through an efficient algorithm based on dynamic programming. It explores all phone sequences within the CN, all their segmentation and all choices of words and scores them with a log-linear model including three features: a word-based LM, the CN posterior probability, and a word penalty.

Efficiency of the CN decoder is achieved by fast prefetching phone transcriptions, through a prefix tree representation of the phrase table [14]. Finally, notice that the same decoder can be used to translate both texts and CNs, once texts are seen as CNs of fixed depth one.
6. Experiments

In this section we present preliminary results with the phone-based ASR system. In particular, we focused our investigation on the following two aspects of the phone-based ASR system: the use of different CN interfaces and the use of language models of increasing size and vocabulary. In the following, we illustrate the kind of CNs, LMs and phrase-tables we prepared.

We computed CNs according to the two approaches described in Section 4. Namely, CNs (\text{cn-cm}) generated by augmenting the best phone string with information from the phone error confusion matrix, and CNs (\text{cn-wg}) extracted directly from the phone graphs.

In addition, three 4-gram LMs were estimated on the 606M word news corpus, by applying the Improved Kneser-Ney smoothing method \cite{KneserNey} and different pruning strategies: LM \text{lm1} is obtained by filtering out \text{n-grams} with words not among the top frequent 64K words, and all 3-gram and 4-gram occurring once; LM \text{lm2} was computed by only pruning singleton 3- and 4-grams; LM \text{lm3} finally contains the full dictionary and \text{n-grams}. LMs \text{lm2} and \text{lm3} result in a dictionary of about 1.2M words. LM \text{lm3} was created just to test the capability of the phone-based decoder to handle huge amount of \text{n-grams}, and not to increase recognition quality. Indeed, it contains many more \text{n-grams}, but extremely rare. Statistics of the three LMs are reported in Table 2, namely the amount of 4-grams, the perplexity and the OOV rate of both dev and test.

By increasing the vocabulary size, from 64K words to 1.2M words, we observe interesting reductions in OOV rate, from 1.53% to 0.07%. In terms of perplexity, \text{lm2} is much better than \text{lm1} although the amount of 4-grams is very close; viceversa, the addition of singletons to \text{lm3} gives a negligible improvement over \text{lm2} as expected.

Two phrase tables \text{pt1} and \text{pt2} were built by automatically transcribing all words occurring in the 64K and 1.2M word dictionaries, respectively.

For the sake of comparison and to have an upper-bound of performance, we also performed phone-to-word decoding with the best automatic phone transcriptions \text{bst}, and with the phone transcription of the human references \text{ref}. The latter error rate gives a measure of the phone-to-word translation difficulty of the language/task at hand. Errors may in fact occur due to multiple ambiguous segmentations of the phone string and, for a given segmentation, to homophones that can make the word classification ambiguous \cite{Smoothing}.

As a further reference, we also run the FBK-irst recognizer \cite{FBK}, a state-of-the-art word-based ASR system (see Section 2) exploiting a LM comparable with \text{lm1}. On the test set, the FBK-irst recognizer achieves WERs of 11.18% and 13.18%, respectively, on the dev and test set. The Real Time Ratio (RTR) of this system is approximately equal to 6.

The experiments carried out show that the proposed approach is a promising alternative to the conventional word-based decoder. The most important feature is that it can be gracefully scaled up by applying larger language models and wider vocabularies. By increasing the vocabulary size from 64K words to 1.2M words (\text{lm1} vs. \text{lm2}) we observe consistent and significant WER reductions. Instead, WER differences are negligible by increasing the number of n-grams (\text{lm2} vs. \text{lm3}). By comparing these results with the values of Table 2, it is evident the strong correlation between LM perplexity and WER.

Henceforth, this suggests that performance of our phone-based decoder could be further improved by exploiting better LM, no matter of their size.

The use of the larger phrase table \text{pt2}, corresponding to the 1.2M word dictionary, causes a significant increment of the average decoding time per utterance. This is because many more alternative phone segmentations can be exploited due to the larger phrase table. Decoding takes more time with the CNs, because they store many phone strings. Instead, larger amount of n-grams does not affect decoding time.

Using CNs as interface gives significantly better results than using single phone transcription hypotheses. Experiments show that CN decoding allows to reduce WER by about 28% with respect to 1-best string decoding. CNs generated from the phone graphs \text{cn-wg} are definitely better than those computed from the phone error matrix \text{cn-cm}. In fact, the former are extremely richer than the latter (2.26 vs 5.10 Ph GER), are more compact (1.88 vs. 2.58 of average depth per utterance), and permit to achieve substantially smaller WERs (16.08 vs. 20.97). We explain the lower quality of \text{cn-cm} with the fact that the posterior probabilities extracted from the phone confusion matrix are independent from the utterance at hand.

Another remarkable feature of the new approach is speed. Even in the slowest case, namely decoding with \text{cn-wg} CNs and the \text{lm2-pt2} LM, the two decoding steps have jointly a total RTR close to 3, while the RTR of the reference word-based ASR system is 6. In this system's configuration, time splits equally between phone decoding e generation of the CNs and CN decoding.

7. Conclusions and Future Work

In this paper a two stage ASR architecture was presented: the proposed system couples a weakly constrained phone recognizer with a phone-to-word decoder able to efficiently decode CNs. The two-step approach has several advantages, in particular it allows the use of very large LMs and huge dictionaries and it is significantly faster than word-based ASR.

Results achieved on transcription of speeches of the Italian parliament show that the proposed approach is a valid alternative to word-based ASR, particularly for SDR application where a large vocabulary and a consequently low OOV rate is a desirable feature.

WER achieved with the two-step approach is still worse than the one achieved by the reference word-based ASR system. However, performance improvement are expected by using huge LMs and by improving the process that generates the phone CNs. In particular, better pruning strategies should be
Table 3: Recognition results (WER %) for different input sources and different configurations of the CN decoder. Phone Graph Error Rate and average depth of the input sources and the average decoding time per utterance are also reported.

<table>
<thead>
<tr>
<th>Input</th>
<th>PhGER (%)</th>
<th>avg depth</th>
<th>configuration</th>
<th>WER (%) [0.95% confidence interval]</th>
<th>avg dec time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
<td>test</td>
<td>dev</td>
<td>test</td>
<td>dev</td>
</tr>
<tr>
<td>ref</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>4.92</td>
</tr>
<tr>
<td>lst</td>
<td>6.34</td>
<td>7.03</td>
<td>1.00</td>
<td>1.00</td>
<td>23.90</td>
</tr>
</tbody>
</table>

Future work will investigate the impact of the proposed approach in several topics of interest in the ASR community. Open vocabulary ASR, for example, can greatly benefit from the proposed two step decoder. A particularly interesting aspect we would like to investigate is the use of multiple LMs during decoding and the use of a subword-based recognizer with the aim of detecting and recognizing OOV words.

Other interesting applications that will be investigated in future works are system combination and cross-system adaptation. With these techniques the output of the two-pass system may be used to improve the performance of the reference baseline system, for example by combining the outputs by means of techniques like ROVER.

8. Acknowledgements

This work has been partially funded by the European Union under the project Judicial Management by Digital Libraries Semantics (JUMAS FP7-214306) and by the Autonomous Province of Trento (Italy) under the project RIVA.

9. References


