INVESTIGATIONS ON EXEMPLAR-BASED FEATURES FOR SPEECH RECOGNITION TOWARDS THOUSANDS OF HOURS OF UNSUPERVISED, NOISY DATA

Georg Heigold, Patrick Nguyen, Mitchel Weintraub, and Vincent Vanhoucke

Google Inc., USA

ABSTRACT

The acoustic models in state-of-the-art speech recognition systems are based on phones in context that are represented by hidden Markov models. This modeling approach may be limited in that it is hard to incorporate long-span acoustic context. Exemplar-based approaches are an attractive alternative, in particular if massive data and computational power are available. Yet, most of the data at Google are unsupervised and noisy. This paper investigates an exemplar-based approach under this yet not well understood data regime. A log-linear rescoring framework is used to combine the exemplar-based features on the word level with the first-pass model. This approach guarantees at least baseline performance and focuses on the refined modeling of words with sufficient data. Experimental results for the Voice Search and the YouTube tasks are presented.

Index Terms— Exemplar-based speech recognition, conditional random fields, speech recognition

1. INTRODUCTION

State-of-the-art speech recognition systems are based on hidden Markov models (HMMs) to represent phones in context. These models are convenient due to their simplicity and compactness. However, it is hard to incorporate long-span acoustic context into this type of models, without pooling observations from different examples on the frame level.

Non-parametric, exemplar-based approaches such as k-nearest neighbors (kNN) appear to be an attractive alternative to overcome this limitation of conventional HMMs and may be more effective at capturing the large variability of speech. In this paper, we investigate an exemplar-based (also known as template-based) rescoring approach to speech recognition, which can be considered a variant of kNN on (pre-)segmented acoustic units such as words.

Like for most non-parametric approaches, the main concerns about exemplar-based speech recognition are that it requires large amounts of data and thus, massive computational power. The origin of the complexity is twofold. First, there is no compact representation as in case of conventional HMMs and all data need to be memorized and processed. Second, the Dynamic Time Warping (DTW) distance [1, 2] is used to measure the similarity between two templates. Using dynamic programming, the computation of this distance has quadratic complexity in the length of the templates.

Current distributed computation and storage systems can process an unprecedented volume of speech data collected from mobile and video sharing speech applications. However, most of the data are of low quality in the sense that it is unsupervised and noisy data. For these reasons, the focus of this paper shall be on the following two issues.

- Investigate exemplar-based speech recognition for thousands of hours of unsupervised and noisy data.
- As a preliminary, implement an infrastructure that makes such investigations feasible and reasonable.

To the best of the authors' knowledge, existing work has focused on the TIMIT and Wall Street Journal (WSJ) tasks with comparably little (less than 100 hours) and clean data [3, 2, 4, 5]. Results for the Voice Search task for a few hundred hours of supervised data were shown in [6], although heavy template selection was used to keep the computation time sufficiently low.

In this paper, we investigate exemplar-based features on the word level. These features are combined with the first-pass model in a rescoring pass. For the combination, we use conditional random fields on n-best lists/word lattices [7]. This approach has the advantage that additional information can be incorporated without building a competitive, stand-alone template system like for conventional system combination using ROVER or confusion network combination [8]. Unlike for a stand-alone template system [2], full coverage of the data by the templates is not required because a first-pass model is used as a background model. In particular, this allows us to use the word as the acoustic unit. This choice not only increases the acoustic context compared to the conventional triphones (except for short words) but also helps to structure the search space for faster search.

The remainder of the paper is organized as follows. Section 2 defines the template features to be used in this work. Section 3 discusses how these features are used for rescoring a first-pass model. Experimental results are provided in Section 4. The paper is concluded in Section 5.

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

2. EXEMPLAR-BASED FEATURES

This section describes the template features used in this paper. These features use the words as the acoustic unit. Here, we distinguish between the “1 feature / word” and the “1 feature / template” approach. The first approach is similar to the template features used in [6, 4] and are inspired by the exemplar-based approach to speech recognition [1, 2]. The second approach resembles a radial basis function kernel on the word level. It is expected to be less sensitive to errors than the first approach because there is no explicit notion of correct and incorrect templates.

The template features are based on some distance $d(X, Y)$ between two sequences of feature vectors X, Y of different length, say the common 39-dimensional PLP features. A common metric is the Dynamic Time Warping (DTW) distance [1, 2]. The simplest variant is defined as the summed Euclidean distances of the best warping of the two sequences. The warping usually is subject to certain constraints, for example, monotonicity and bounded jump size. The DTW distance can be computed by dynamic programming, with a complexity quadratic in the number of frames. The distance is length-normalized for further processing to make templates of different length comparable.

Two possible definitions for template features follow. They are based on the segmented word W (for example, a word lattice arc) and the frame-features X associated with this word segment.

“1 feature / word” In this approach, the feature is set to the average distance between the hypothesis X and the k-nearest templates of X associated with the hypothesis word W, $Y \in \mathsf{KNN}_W(X)$ if the word hypothesis W' matches the template word W. Otherwise, it is set to zero.

$$f_{W'}^{tmpl}(X, W) = \begin{cases} \sum\limits_{Y \in \mathsf{KNN}_W(X)} \frac{d(X,Y)}{|\mathsf{KNN}_W(X)|} & \text{if } W' = W \\ 0 & \text{otherwise} \end{cases}$$

There is one feature for each word. Only one feature can be active at the same time. These features require that the templates are correctly classified. This type of features was previously used on the utterance level [6] and for word templates composed of phone templates [4]. A similar approach for isolated word recognition was presented in [3].

“1 feature / template” In this approach, the DTW distances are directly used as the features. Considering all templates for each word is not feasible at our scale. So we only activate the templates for the word under consideration. Thus, there is one feature for each template, resulting in several millions of features. Furthermore, the distances are exponentiated to achieve a more sparse representation and thus faster training. In addition, this non-linearity allows us to model arbitrary decision boundaries and not only (piecewise) quadratic decision boundaries as in the absence of this non-linearity. In the “1 feature / word” approach, this non-linearity is not essential due to selecting the k-nearest neighbors for averaging.

$$f_Y^{kernel}(X, W) = \begin{cases} \exp(-\beta d(X, Y)) & \text{if } Y \text{ template of } W \\ 0 & \text{otherwise} \end{cases}$$

The scaling factor β needs to be tuned. Note that there is no explicit notion about correct and incorrect templates and the learning algorithm hopefully learns the relevance of each template. This appears to be an attractive property in our case.

M-gram word templates are word templates in an m-gram word context. Note that ‘m-gram’ only refers to the acoustic context but not the acoustic unit which remains the word. For example, bigram word templates are word templates that are preceded by a certain word. M-gram word templates are the same as m-gram word unit templates except that they “pinch” the search space for DTW at the word boundaries and thus, do not increase the complexity compared to word templates. In particular for short words, they may be helpful to take account of co-articulation effects.

There are two good reasons for using word template features. First, they can span a longer acoustic context than conventional triphones. This may be helpful, in particular in combination with large amounts of data. Second, this choice helps to structure the search space for DTW. This is an essential issue if dealing with thousands of hours of data and probably is a natural and simple way to implement search trees.

3. COMBINATION & OPTIMIZATION

The combination of the template features with the first-pass model is done with a segmental conditional random field (SCARF) [7]. This is a conditional random field defined on word lattices. Thus, the training is basically the same as for lattice-based discriminative training of HMMs, see for example [9].

The features of the conditional random field are defined on the word arc level, see Section 2 for examples. In addition to the template features, the language and acoustic model scores are used as features. This guarantees that the rescored model performs no worse than the first-pass baseline model.

The optimal model weights are determined by *a posteriori* estimation using a Gaussian and a Laplace prior [10]. The resulting training criterion is also known as maximum mutual information (MMI) with ℓ_2- and ℓ_1-regularization. ℓ_1-regularization is used for implicit feature selection. This is helpful because only a small fraction of the millions of template features are expected to carry additional information. The optimization is done with the general-purpose algorithms L-BFGS or Rprop.

4. EXPERIMENTAL RESULTS

The template features are evaluated on the Voice Search and the YouTube audio transcription tasks, both for US English.

Table 1. Training data statistics for Voice Search and YouTube.

Task	[h]	#utterances	#words	supervised
Voice Search	3k	3.3M	11.2M	60%
YouTube	4k	4.4M	40M	0%

4.1. Tasks & Data

The training data for Voice Search consist of supervised and unsupervised data. Roughly half of the training data are supervised, see Table 1. The transcriptions for the unsupervised data were generated by decoding using the first-pass model trained on the supervised portion of the data. Only utterances with an n-best list containing the oracle are used for training.

The training data for YouTube are all unsupervised, see Table 1. The transcriptions are obtained by synchronizing the transcriptions the users uploaded with the audio data. There is some heuristics to filter out utterances with a bad transcription. All utterances are used for training. To avoid problems with vanishing probabilities in MMI, the reference is set to the hypothesis in the lattice with the lowest edit cost. The oracle word error rate is 29%, compared to 45% for the single best.

All word hypotheses from the reference are taken as templates but not more than 5k (random) templates per word. This is a rather conservative threshold and mainly helps the top words not to have millions of templates. In case of YouTube, we only select from the word hypotheses in the reference that align with the transcription with zero edit cost. Due to the high oracle error rate, this reduces the number of templates significantly. As expected, this filtering seems to be important for the "1 feature / word" but not so much for the "1 feature / template" approach, see Section 2. For better comparison, however, we use the same filtered templates for all experiments. With this template selection, we have roughly 4M out of 10M training tokens for Voice Search and up to 20M out of 40M training tokens for YouTube (of which 20M tokens align with non-zero edit cost). This amounts to at least 1k hours of templates for either task. The coverage of the test data by templates is pretty good. 98%/90% (Voice Search) and 90%/85% (YouTube) of the test data is covered by 10/100 or more templates.

The first-pass acoustic model is a conventional, discriminatively trained HMM using Gaussian mixtures and PLPs as the front-end features. The decoding uses a trigram language model. The vocabulary size for decoding is 1M (of which 100k are seen in the training or test data) for Voice Search and 127k for YouTube.

4.2. Preliminary Analysis

We analyzed different aspects of the template features described in Section 2 for development and tuning. The templates use the same front-end features as the first-pass acoustic model (PLPs). In case of YouTube, they include CMLLR. Rprop and L-BFGS were used for optimization of Voice Search and YouTube, respectively.

Table 2. Performance of template features with unigram language model (LM) in comparison with first-pass acoustic model ('AM') for YouTube.

Features	WER [%]
unigram LM + AM	66.1
unigram LM + "1 feature / word"	70.4
unigram LM + "1 feature / template"	63.8

Table 3. "1 feature / word" vs. "1 feature / template" word template features on top of first-pass model ('AMLM').

Features	WER [%]	
	Voice Search	YouTube
AMLM	14.7	57.0
+ "1 feature / word"	14.3	56.7
+ "1 feature / template"	14.1	55.9

Quality of template features. To gauge the template features in comparison with the first-pass acoustic model, models with only the first-pass acoustic model score or the template features are trained. To reduce the effect of implicitly training unigram contexts in case of template features, the unigram language model score is added. Note that around 10% of the test data are not covered by word templates and thus, are likely to be misclassified. The results in Table 2 suggest that the template features are competitive with the first-pass acoustic model within our rescoring approach.

"1 feature / word" vs. "1 feature / template." Table 3 compares the "1 feature / word" and the "1 feature / template" approach. The "1 feature / template" approach seems to outperform the "1 feature / word" approach, see also Table 2. The reason for this may be that the "1 feature / template" approach is able to learn the relevance of each template and is less sensitive to erroneous data.

Effect of data sharpening. Data sharpening is a preprocessing step known from the k-nearest neighbors (kNN) approach. It is used for outlier correction. In our case, the data sharpening replaces the frame-features with the average over the k-nearest features aligned to the same triphone. Triphones are chosen mainly to make the underlying kNN problem feasible. These features are then used for the templates. Substantial gains were shown from data sharpening for exemplar-based speech recognition [2]. However, it is not obvious if this technique is also effective in our combination approach and for large amounts of data. Table 4 summarizes the comparative results. There is a clear gain if we do kNN on the segmented words and only consider reference hypotheses that are in the lattice ('kNN, with oracle'). However, this gain appears less optimistic in the context of all word hypotheses ('kNN, all') and almost completely vanishes after the combination with the first-pass model.

Bigram word templates. By default, the maximum number of templates per word is set to 5k. This value for the threshold

Table 4. Effect of data sharpening under different conditions for YouTube.

	WER [%]	
	data sharpening	
Setup	no	yes
kNN, with oracle	26.1	20.4
all	62.4	59.5
AMLM + "1 feature / template"	56.4	55.9

Table 5. Unigram vs. bigram word template features for YouTube, on top of first-pass model ('AMLM').

Features	WER [%]
AMLM + "1 feature / template" (unigram)	55.9
AMLM + "1 feature / template" (bigram)	55.0

makes the computation reasonably efficient. However, does this also imply optimal performance? Internal tests suggest that increasing the number of templates per word does not help. Increasing the acoustic context appears to be a more attractive approach to use more data (doubles the amount of templates to 2k hours), without substantially increasing the complexity. In addition, bigram word templates increase the acoustic context and take into account co-articulation effects to some degree. This is expected to be useful for YouTube where short words dominate: the ten most frequent words are 'the', 'to', 'and', 'a', 'you', 'of', 'that', 'is', 'in', 'it', and make up one third of the seen words. The results are shown in Table 5.

4.3. Combination Results for Template Features

Table 6 shows the combination results for the template features. The results are on top of an MMI-trained first-pass model, which reduces the gain by the effect of discriminative training. The observed gains are modest for Voice Search and small for YouTube. Voice Search probably performs better because a substantial portion of the training data is supervised whereas there is no supervised training data available for YouTube.

Interestingly, only a few percent of the templates in the "1 feature / template" approach are active for optimal ℓ_1-regularization. This observation can be used for a simple speed-up of the rescoring pass by a factor of ten for example. This, however, does not affect the training time which for pre-computed template features, is comparable to that of conventional MMI training but using roughly ten times more iterations to converge. Pre-computing the template features is almost as expensive as the training itself.

Table 6. Template features combined with the first-pass model ('AMLM'), unigram word templates for Voice Search and bigram word templates for YouTube.

Features	WER [%]	
	Voice Search	YouTube
AMLM	14.7	57.0
+ "1 feature / template"	14.1	55.0

5. SUMMARY

In this paper, we investigated exemplar-based features for large-scale speech recognition. These features were combined with the first-pass model using a lattice-based, rescoring approach. Results for up to 20 million word templates drawn from thousands of hours of unsupervised and noisy training data were shown. The currently observed gains probably are insufficient to justify the increased complexity. In our opinion, the most likely reason for the negative result is the hard but for many applications realistic data conditions. More speculations may include the high error rates in case of YouTube, the MMI baseline that reduces the gain by the discriminative training, or simply that the template features are not complimentary enough.

References

[1] O. Ghitza and M. Mohan Sondhi, "Hidden Markov models with templates as non-stationary states: an application to speech recognition," *Computer Speech and Language*, vol. 2, pp. 101–119, 1993.

[2] M. De Wachter et al., "Template-based continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 15, no. 4, pp. 1377–1390, 2007.

[3] S. Axelrod and B. Maison, "Combination of hidden Markov models with dynamic time warping for speech recognition," in *Proc. ICASSP*, Montreal, Quebec, Canada, May 2004, pp. 173–176.

[4] G. Zweig et al., "Speech recognition with segmental conditional random fields: A summary of the JHU CSLP 2010 summer workshop," in *Proc. ICASSP*, Prague, Czech Republic, May 2011, pp. 5044–5047.

[5] T.N. Sainath et al., "Exemplar-based sparse representation features: From TIMIT to LVCSR," *IEEE Transactions on Speech and Audio Processing*, vol. 19, no. 8, pp. 2598–2613, 2011.

[6] G. Heigold et al., "A flat direct model for speech recognition," in *Proc. ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 3861–3864.

[7] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *Proc. ASRU*, Merano, Italy, Dec. 2009, pp. 152–157.

[8] B. Hoffmeister et al., "Cross-site and intra-site ASR system combination: Comparisons on lattice and 1-best methods," in *Proc. ICASSP*, Honolulu, HI, USA, Apr. 2007, pp. 1145–1148.

[9] G. Heigold, R. Schlüter, and H. Ney, "Modified MPE/MMI in a transducer-based framework," in *Proc. ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 3749–3752.

[10] S. Chen and R. Rosenfeld, "A Gaussian prior for smoothing maximum entropy models," Technical Report CMUCS-99-108, Computer Science Department, Carnegie Mellon University, 1999.