Comparison of Discriminative Models for Lexicon Optimization for ASR of Agglutinative Language

MIJIT ABLIMIT[†] TATSUYA KAWAHARA[†]
ASKAR HAMDULLA[††]

For automatic speech recognition (ASR) of agglutinative languages, selection of lexical unit is not obvious. Morpheme unit is usually adopted to ensure the sufficient coverage, but many morphemes are short, resulting in weak constraints and possible confusions. We have proposed a discriminative approach to select lexical entries which will directly contribute to ASR error reduction, considering not only linguistic constraint but also acoustic-phonetic confusability. It is based on an evaluation function for each word defined by a set of features and their weights, which are optimized by the difference of word error rates (WERs) by the morpheme-based model and those by the word-based model. In this paper, we investigate several discriminative models to realize this scheme. Specifically, we implement with Support Vector Machines (SVM) and Logistic Regression (LR) model as well as simple perceptron. Experimental evaluations on Uyghur LVCSR show that SVM and LR are more robustly trained and SVM results in the best performance with a large dimension of features.

1. Introduction

In agglutinative languages, selection of lexical unit is not obvious and one of the important issues in designing language model for automatic speech recognition (ASR). There is a trade-off between word unit and morpheme unit; generally the word unit provides better linguistic constraint, but increases the vocabulary size explosively, causing OOV (out-of-vocabulary) and data sparseness problems in language modeling. Therefore, the morpheme unit is conventionally adopted in many agglutinative languages, such as Japanese [1], Korean [5], and Turkish [9]. However, most of morphemes are short, often consisting of one or two phonemes, thus they are more likely to be confused in ASR than the word unit. The goal of this study is to incorporate effective word (or sub-word) entries selectively while maintaining the high coverage of the morpheme unit.

There are a number of previous works addressed on this problem, and many of them are based on statistical measures, such as co-occurrence frequency, mutual information, and likelihood [4]-[9]. However, these criteria are not directly related to WER (word error rate). They do not consider phonetic similarity and unit length which are potentially related with confusability in ASR.

We have proposed a novel discriminative approach to select word (or sub-word) entries which are likely to reduce the WER [13]. It is realized by aligning and comparing the ASR results by the morpheme-based model with those by the word-based model. We describe each word by a set of features, and define an evaluation function with their weights. Then, the weights are learned to select "critical" word entries which generate different (probably correct) hypotheses from the morpheme-based units. This learning mechanism is applicable to any unseen words, or even sub-words.

In our previous research, the scheme is realized with a simple perceptron algorithm [13]. In this paper, we investigate more sophisticated models including Support Vector Machines (SVM) and Logistic Regression (LR) model [14]. The proposed method is applied to and evaluated in a large-vocabulary Uyghur ASR system.

2. Overview of the proposed scheme

Overview of the proposed scheme is depicted in Figure 1. Baseline ASR systems are prepared with both morpheme-based units and word-based units, and they are applied to a large-scale speech database. We can use the speech database used for acoustic model training, though it produces the closed recognition results. We can use even un-transcribed speech data, as the proposed learning scheme is realized in an unsupervised manner.

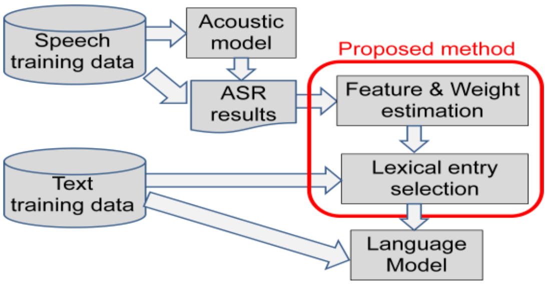

Figure 1: *Overall flow of the proposed training scheme.*

The ASR results by the morpheme-based model and the word-based model are aligned by each word with corresponding morpheme sequences. We assume each word is composed of one or more morphemes, and morpheme units do not cross word boundaries. An example is given in Table 1.

When these two ASR results are different, neither of them is correct in most cases (approximately 68% in our data set). In the majority of the remaining cases, however, the word-based model gives correct hypotheses while the morpheme-based model does not (28.5% vs. 3.5%). Therefore, a naïve method would be to pick up these "critical" word entries (e.g. "cheghinglarda" in the example of Table 1) to be added to the lexicon. When conducted in the closed test-set, it would result in a drastic improvement in

[†] School of Informatics, Kyoto University, Kyoto, Japan.
[††] Institute of Information Engineering, Xinjiang University, Urumqi, China

ASR. However, the method heavily depends on the training data set since it can select only entries observed there, and thus may not have a generality.

Therefore, we introduce a more generalized scheme; we describe each word by a set of features and weights, and optimize the weights to select word entries which give different ASR result and will contribute to WER reduction. Once these weights are learned, we can apply the resultant evaluation function to any word in the text training database to determine whether or not it should be included in the lexicon. Based on the lexicon which is based on the morpheme unit and enhanced with effective word units, the final language model is trained.

Table 1. *Example of ASR results of morpheme and word units*.

Reference word	Yash cheghinglarda bilim elishinglar kerək
Reference morph	Yash chegh_ing_lar_da bilim el_ish_ing_lar kerək
word ASR result	Yash cheghinglarda bilim berishinglar kerək O O O X O
morph ASR result	Yash chegh_ing_da bilim el_ish_ing_lar kerək O X O O O

3. Comparison of discriminative models

In the proposed scheme, each word is described by a set of features $(x_{i1}, x_{i2}, \dots x_{in})$ of the constitute morphemes ($m_1 m_2$ ...), and its desired value y_i defined by the differences of ASR results of two units. We assume that they are binary (1 for true, 0 or -1 for otherwise). Given all the training pairs $(\boldsymbol{x_i}, y_i), i = 1, \dots l,\ \boldsymbol{x_i} \epsilon \{0,1\}, y_i \epsilon \{-1,+1\}$, we feed them to the training scheme. In this work, we adopt and compare three different machine learning algorithms: perceptron, SVM, and LR.

For the perceptron algorithm, we can define an evaluation function as a linear weighted sum of the features [13].

$$f(w) = \sum_s (x_{is}\, w_{is}) = \boldsymbol{x_i}\, \boldsymbol{w_i} \tag{1}$$

The standard sigmoid function is introduced to map the above evaluation score to the 0-1 range.

$$g(w) = \frac{1}{1+e^{-f(w)}} \tag{2}$$

$$g'(w)|_{f(w)} = g(w)(1 - g(w)) \tag{3}$$

Then, the weight vector is updated as:

$$\boldsymbol{w} = \boldsymbol{w} + \eta\, g'(w)(y_i - g(w))\boldsymbol{x_i} \tag{4}$$

The learning rate parameter η is adjusted at every iteration to prevent excessive fluctuation. Here we simply reduce it by a factor of 10. This learning converges in several iterations, and we terminate at the third iteration in the experiments.

For the SVM and LR, we adopt a linear binary classifier [14]. Given the same set of training sample pairs $(\boldsymbol{x_i}, y_i)$, both methods solve the following unconstrained optimization problem with different loss functions $\xi(w;\ x_i, y_i)$:

$$min_w \frac{1}{2}\, w^T w + C \sum_{i=1}^{l} \xi(w;\ x_i, y_i) \tag{5}$$

where C>0 is a penalty parameter. For SVM, the two popular loss functions are:

$$\max(1 - y_i w^T x_i, 0) \tag{6}$$

and

$$\max(1 - y_i w^T x_i, 0)^2 \tag{7}$$

The former is referred to as L1-SVM, and the latter is L2-SVM. In our experiments, we use L2-SVM.

The loss function for LR is:

$$\log(1 + e^{-y_i w^T x_i}) \tag{8}$$

which is derived from a probabilistic model. The SVM optimization is stopped at the tolerance of 0.1, and the LR training stopped at tolerance of 0.001.

The training feature sample pairs $(\boldsymbol{x_i}, y_i)$ are extracted independently for every word and its corresponding morpheme sequence. When the word is misrepresented by the morpheme sequence, the desired value is $y_i = +1$, otherwise $y_i = -1$. These models estimate every word according to its features $\boldsymbol{x_i}$, which indicates the potential importance of the word to be included in the lexicon, or how likely WER will be reduced by adding this word entry. Note that these models can be used for any words or even sub-words consisting of morpheme sequences, so that we can select effective entries which would not be correctly recognized by the morpheme-based model.

3.1 Weight estimation with discriminative learning

The values of the weights $\boldsymbol{w} = \{w_s\}$ are estimated based on the above described models using the training data set. The desired output y_i is defined as binary, corresponding to the CRITICAL_CASE in which the word-based model outputs a different hypothesis from the sequence generated by the morpheme-based model.

$$y_i = \begin{cases} +1 & \text{if CRITICAL\_CASE is true} \\ -1 & \text{otherwise} \end{cases}$$

Note that the above judgment does not refer the correct hypotheses. There are some cases in which the word-based model makes an error while the morpheme-based model generates a correct hypothesis as shown in the right-hand example of Table 3. However, the ratio of such cases among all differences is only 3.5% as shown in the previous section. We also introduce sample filtering as described in the next sub-section. The property of not using the reference transcripts makes the proposed training in an unsupervised fashion, so that we can make use of enormous un-transcribed speech data.

3.2 Filtering training samples

We introduce filtering of training samples so that only reliable samples are fed to the training. Specifically, we selectively use the samples whose frequency of CRITICAL_CASE is more than N times over the entire training data set. This will also be effective for discarding erroneous samples made by the word-based model, as discussed in the previous sub-section.

3.3 Lexical features

In our previous paper [13], we investigated a variety of lexical features considered for the proposed scheme, and found the morpheme N-gram features are most effective though it makes a large dimension. Thus, we adopt them in this work. We describe the candidate word as “word”, and the corresponding morphemes as “m_i”. A specific weight w is estimated for each unigram or bigram entry.

$$\mathrm{x}_{\mathrm{unigram}\_m_i} = \begin{cases} 1 & \text{if } morph.\, m_i\ exists\ in\ word \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{x}_{\mathrm{bigram}\_m_i\, m_j} = \begin{cases} 1 & \text{if } morph.\, bigram\ (m_i\ m_j)\ exists\ in\ word \\ 0 & \text{otherwise} \end{cases}$$

Below is an example of the bigram morpheme sequence appeared in Table 1.

$$\mathrm{x}_{\mathrm{bigram}}(\_ing\ \_lar) = 1$$

3.4 Lexicon design

These features are then generalized to all words in the text corpus for language model training. If the candidate word is judged as CRITICAL_CASE or the evaluation function $g(w)$ is larger than a threshold (=0.5), we select it to be included in the lexicon. Otherwise the word is left as morpheme units.

Furthermore, the method can be applied to sub-words, which are composed of morpheme sequences within a word. Specifically, we try to search for sub-word entries that satisfy the lexical features. The search is exhaustively done from the beginning of all words by concatenating the following morphemes while the above-mentioned condition is met. If the condition is not met, the search is re-started there.

4. Experimental evaluations

The method has been implemented and applied to our Uyghur LVCSR system. A speech corpus of general topics is prepared to build an acoustic model of Uyghur. This corpus is also used as the training data set for lexicon optimization addressed in this work. A held-out test data set is prepared from readings of newspaper articles. Specifications of the data sets are summarized in Table 2.

Table 2. *Statistics of speech corpora.*

corpus	sentences	persons	total utterances	time (hour)
training	13.7K	353	62K	158.6
test	550	23	1468	2.4

Acoustic models based on tri-phone HMMs with 3000 shared states and 16 Gaussian mixtures are trained for 34 Uyghur phones (8 vowels, 24 consonants, and 2 silence models). Acoustic features consist of 12 MFCCs, ΔMFCCs and $\Delta\Delta$MFCCs together with Δpower and $\Delta\Delta$power.

For language modeling, a text corpus of 630K sentences is collected over general topics from newspaper articles, novels, and science textbooks. The sentences are segmented to morpheme and word units by our morphological analyzer [3].

Two different lexical units (word and morpheme) are used to build n-gram (3-gram and 4-gram) language models. In this work, the Kneser-Ney smoothing method is adopted. The best performance by the baseline models are WER=25.77% by the word-based 3-gram model with a lexicon size of 230K, and WER=28.11% by the morpheme-based 4-gram model with a lexicon size of 27K. Once the lexicon is enhanced by adding the word or sub-word entries, 4-gram language model is trained again.

4.1 Effect of sample filtering

We investigate the effect of sample filtering described in Section 3.2. In this experiment, we use morpheme unigram features applied to the word level. The WERs obtained by changing the threshold (N) values are listed in Table 3. We can see that removing outlier (possibly erroneous) samples of only one occurrence is effective for the perceptron algorithm, but not so much for the SVM and LR. This results show that SVM and LR are trained more robustly and reliably against outlier samples. Based on the results, we set N=0 for SVM and LR, and N=2 for the perceptron in the following experiments.

Compared with the baseline morpheme-based models (WER=28.11%), all methods lead to significant improvement and the accuracy is comparable to the best word-based model (WER=25.77%). Note that the lexicon size of the enhanced morpheme-based model is much smaller than the word-based model (230K with Cutoff-2), and still expected to give broad coverage.

4.2 Comparison of sub-word and word selection

We also generate sub-word lexical entries by using the morpheme N-gram features. Here we also compare the morpheme unigram and bigram features. The dimension of the unigram features is 17K and that of the bigram is 53K. The result in Table 4 shows that this method reduces both WER and the lexicon size significantly. The proposed optimization is more effective when conducted thoroughly in the sub-word level than the word level. The sub-word-based model trained with the bigram feature outperforms the best word-based model in accuracy with the lexicon size of one fourth. From the results we can see that the SVM and LR methods are more effective especially with a large dimension of the bigram features.

Table 3. *Effect of sample filtering threshold with unigram feature*.

threshold		N=0	N=1	N=2	N=3	N=4	N=5
perceptron	WER (%)	26.69	25.93	**25.87**	26.18	26.28	26.54
	Lexicon size	104.5K	90.2K	74.8K	63.6K	55.3K	50.1K
LR	WER (%)	**25.99**	25.57	25.91	25.93	26.01	26.22
	Lexicon size	102.4K	91.2K	79.9K	70.1K	62.4K	56.5K
SVM	WER (%)	**26.05**	26.03	25.93	25.93	26.00	26.22
	Lexicon size	103.4K	94.6K	83.7K	73.5K	65.4K	59.2K

Table 4. *Comparison of results on different units and features*.

Units		Word		Sub-word	
Features		unigram	bigram	unigram	bigram
perceptron	WER (%)	25.87	25.99	25.96	25.27
	Lexicon size	74.8K	67.3K	40.7K	49.9K
LR	WER (%)	25.99	25.75	25.77	24.87
	Lexicon size	102.4K	85.4K	44.0K	65.8K
SVM	WER (%)	26.05	25.86	27.05	**24.61**
	Lexicon size	103.4K	80.1K	34.7K	**55.1K**

5. Conclusion

We have investigated a novel discriminative approach to lexicon optimization for agglutinative languages. It can take into account not only linguistic constraint but also acoustic-phonetic confusability in ASR, and is directly linked to the improvement of ASR accuracy. We also made comparison of discriminative models of SVM, LR, and perceptron, and found that SVM and LR are more effective than the perceptron algorithm that was previously used. The proposed scheme is realized in an unsupervised manner, so it can be applied to a large amount of un-transcribed speech data.

Reference

1) T. Kawahara et al.:Free software toolkit for Japanese large vocabulary continuous speech recognition, In Proc. ICSLP, Vol.4, pp.476–479, 2000.
2) M. Ablimit, G. Neubig, M. Mimura, S. Mori, T. Kawahara, A. Hamdulla.:Uyghur Morpheme-based Language Models and ASR, In Proc. IEEE-ICSP, 2010.
3) M. Ablimit, M. Eli, and T. Kawahara.: Partly-Supervised Uyghur morpheme segmentation, In Proc. Oriental-COCOSDA Workshop, pp.71–76, 2008.
4) G. Saon, M. Padmanabhan.:Data-Driven Approach to Designing Compound Words for Continuous Speech recognition, IEEE Trans. Speech and Audio Processing, Vol.9, No.4, 2001.
5) O.-W. Kwon and J. Park.:Korean large vocabulary continuous speech recognition with morpheme-based recognition units, Speech Communication, vol. 39, pp. 287–300, 2003.
6) O-W. Kwon.:Performance of LVCSR with morpheme-based and syllable-based recognition units, In Proc. ICASSP, pp.1567–1570, 2000.
7) M. Jongtaveesataporn, I. Thienlikit, C. Wutiwiwatchai, S. Furui.:Lexical units for Thai LVCSR, Speech Communication, pp.379–389, 2009.
8) K. Hacioglu, B. Pellom, T. Ciloglu, O. Ozturk, M. Kurimo, M. Creutz.: On Lexicon Creation for Turkish LVCSR, In Proc. Eurospeech, 2003.
9) Ebru Arisoy, Hasim Sak.:Murat Saraclar. Language Modeling for Automatic Turkish Broadcast News Transcription, Proc. Interspeech, 2007.
10) Brian Roark, Murat Saraclar, and Michael Collins.:Discriminative n-gram language modeling, Computer Speech and Language, 21(2):373–392, 2007.
11) M. Collins, B. Roark, M. Saraclar.:Discriminative syntactic language modeling for speech recognition, In Proc. ACL, pages 507–514, 2005.
12) M. Collins.:Discriminative training methods for HMMs: Theory and experiments with perceptron algorithm, In Proc. EMNLP 2002.
13) M. Ablimit, T. Kawahara, A. Hamdulla.:Discriminative approach to lexical entry selection for Automatic Speech Recognition of agglutinative language, In Proc. ICASSP 2012.
14) R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin.:LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research 9(2008), 1871-1874.

Acknowledgments This work is supported by JSPS Grant-in-Aid-for Scientific Research (KAKENHI) and NCET of China.