論文 / 著書情報
Article / Book Information

Title	A Large Vocabulary Continuous Speech Recognition System for Indonesian Language
Author	Dessi Puji Lestari, Koji Iwano, Sadaoki Furui
Journal/Book name	15th Indonesian Scientific Conference in Japan, Vol. , No. , pp. 17-22
発行日 / Issue date	2006, 8

Combinations of Language Model Adaptation Methods applied to Spontaneous Speech

Luc LUSSIER†, Edward W. D. WHITTAKER†, and Sadaoki FURUI†

† Department of Computer Science, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan

Abstract This paper presents results for combinations of unsupervised language model adaptation methods applied to the CSJ corpus. Data sparsity is a common problem shared by all automatic speech recognition tasks but it is specially acute in the case of spontaneous speech recognition. The method proposed in this paper combines information from two readily available sources, clusters of presentations from the training corpus and the transcription hypothesis, to create word-class n-gram models that are then interpolated with a general language model. Experimental results show that a relative reduction in word error rate of 10.4%, 10.4% and 5.0% is obtained on the three test sets used.

Key words spontaneous speech recognition, unsupervised language model adaptation, clustering

1. Introduction

The performance of state-of-the art automatic speech recognition systems has steadily improved as both new research and a larger amount of data are applied to this challenging task. This is especially true about the difficult spontaneous speech recognition task where the availability of the "Corpus of Spontaneous Japanese" (CSJ) [9] has provided a considerable amount of appropriate training data.

Our current baseline mean word error rate for the CSJ corpus using the 3 test sets described by Kawahara et al. [4] is 26.8%. In order to reduce the word error rate, we propose a method that combines information from the transcription hypothesis as well as from clusters of presentations and interpolate those models with a general language model.

2. Language model adaptation

The language models used to conduct our experiments are based on the combination of a general language model and one or more specialized language models using linear interpolation as illustrated by the following formula:

$$p(w|h) = \lambda_0 \cdot p_g(w|h, T_0) + \sum_{j=1}^{|T|} \lambda_j \cdot p_s(w|h, T_j) \quad (1)$$

where w is the current word for which the probability is calculated, h represents the history, λ_j is the weight attributed to each model such that $\sum \lambda_j = 1$, p_g is the general language model built from the whole training corpus T_0, $|T|$ gives the number of clusters and p_s corresponds to a specialized language model distinguished by its training source T_j.

2.1 Clustering presentations

All the documents found in the CSJ training corpus are grouped into topic clusters. Each cluster, referred to by the symbol T_j where $1 \leqq j \leqq |T|$ and $|T|$ is the number of presentation clusters, contains a certain number of presentations and each presentation is a member of a single cluster such that $T_i \cap T_j = \varnothing \quad \forall i, j, i \neq j$. The entire corpus of training presentations is referred to as T_0.

The clustering method used is a bottom-up, agglomerative type of clustering based on a word co-occurrence metric. It was used in [3], [10] and is based on [11]. The clustering process works according to the following algorithm:

- Place each presentation P in a single cluster.
- Iterate, until only one cluster is left.
  - For each pair of presentation clusters P_i and P_j,

compute the similarity metric S_{ij}.

– Merge the two clusters that have the highest similarity.

To determine how similar are two presentation clusters, the following similarity metric S_{ij} is used:

$$S_{ij} = \sum_{w \in P_i \cap P_j} \frac{N_{ij}}{|P^w|} \times \frac{1}{|P_i| \times |P_j|} \quad (2)$$

where P_i and P_j represent two presentation clusters, $|P^w|$is the number of presentation clusters that contain the word w, $|P_i|$ is the number of unique words in the cluster P_i and N_{ij}, defined as follow:

$$N_{ij} = \sqrt{\frac{N_i + N_j}{N_i \times N_j}} \quad (3)$$

where N_i which represents the number of presentations in the cluster, is a normalization factor used to prevent the development of a single large cluster.

The clustering is based on all the words from each presentation and the sequence of merge operations is preserved so that any number of clusters can be obtained.

2.2 Building models

Two different kinds of n-gram models are used in our experiments, word n-gram models and word-class n-gram models. The general language model is always a word n-gram model but specialized models can be of either type.

When word-class models are used, the word-class definition is built using the word clustering algorithm described by Kneser and Ney [5] to create $|C|$ different word classes where each word is a member of only one class such that $C_i \cap C_j = \varnothing \quad \forall i, j, i \neq j$.

Also, when using word-class models, an extended notation based on a triplet of variables is used to describe the origin of the data used to train each component of the model resulting in the slightly augmented notation illustrated in the following way:

$$p_s(w|h, D, N, W) = p(w|C(w, D), W) \cdot p(C(w, D)|C(h, D), N) \quad (4)$$

where the extra parameters are used to describe the source from which are trained the word-class definition D, the word-class n-gram distribution N and the word-given-class probability W.

Using the notation $(D|N|W)$ to describe each source of training data, the parameter setting used by Yokoyama et al. in [14], [15] would be described by the following tuple: $(T_0|H|H)$ where T_o represents the entire training corpus and H the transcription hypothesis.

3. Experimental conditions

3.1 Acoustic model

The acoustic features used for the experiments are 25 dimensions vectors consisting of 12 MFCC, their delta as well as the delta log energy. All the models used are gender dependent triphone HMMs with 3000 shared states and 16 Gaussian mixtures. Cepstral mean subtraction is also applied to each utterance.

Table 1 shows the number of presentations and how many hours are used to train the acoustic models. The academic only models are used for the first and second test set and models containing both academic and extemporaneous presentations are used for the third test set.

Table 1 Summary of the data used to create the acoustic models

Model	# talks (# hours)	
	Female	Male
Academic only	166 (42)	787 (186)
Academic and extemporaneous	988 (176)	1508 (310)

3.2 Baseline language model

The baseline language model is built from the transcribed content of about 2590 presentations providing almost 7.5 million words of training data with a vocabulary size of 30678 words. Because there are generally no spaces between characters in written Japanese, the concept of word boundary is not clearly defined. Thus, as defined by Shinozaki and Furui [12], the term word refers to a Japanese morpheme, that is an arbitrary amount of characters, extracted by a morphological analyzer developed by Uchimoto et al. [13] for the CSJ corpus.

All of the training data was used to build a forward word bigram and a reverse word trigram as needed by the Julius speech recognition engine. This baseline language model is referred to as the general language model (G-LM). A variation of the smoothing technique developed by Kneser and Ney introduced in [2] is used with all language models.

3.3 Development and evaluation sets

The first of the three test sets defined in the CSJ

benchmark paper by Kawahara et al. [4] is used as a development set and the last two as evaluation sets. Each test set contains 10 presentations, test set one and two contain only academic presentations while test set three is made of extemporaneous presentations. In addition, test set one contains only presentations made by male speakers whereas test set two and three contain both female and male speakers in equal proportion. The number of words in test sets 1 and 2 are similar but test set 3 is smaller and contains slightly less than 66% of either test set 1 or 2 as summarized in Table 2.

Table 2 Total number of words per test set

Test set number	Number of words
1 (dev)	26515
2	26923
3	17213

3.4 Recognition engine

The recognition is performed with the Julius speech recognition engine version 3.3p3 developed by Lee et al. [6]. In order to accommodate various combinations of word and word-class models, Julius was slightly modified such that language model probabilities could be obtained from an external library.

3.5 Language modelling tools

All the language models used are built with an extended set of tools originally based on the CMU language modelling toolkit [1].

4. Experimental results

4.1 Results on the development set

We first investigated the relationship between adaptation on a per-utterance and per-presentation basis. Tables 3, 4 and 5 show the word error rate for all combinations of $|T|$ equal to 1, 4, 8 and 16 presentation clusters and word n-grams or $|C|$ word-classes equal to 258, 514 and 1026 word-class n-grams. All word-class models are built with the $(T_j|T_j|T_j)$ parameter setting.

It was found in [7] that the word error rate obtained when combining the general language model (G-LM) with a single best word-class model (WC) with a fixed ratio was giving inferior results compared to when all the available models were assigned an interpolation weight with the EM algorithm. The results for tests where the single best word-class model giving the lowest perplexity value on a single transcribed utterance is chosen are

Table 3 Average word error rate (%) for test set 1, combining the general language model with the best model, based on per utterance perplexity, with a fixed ratio of 65:35

$\|T\|$	Word model	$\|C\|$ 258	$\|C\|$ 514	$\|C\|$ 1026
1	27.67	27.22	27.36	27.56
4	27.15	26.94	26.98	27.15
8	26.92	26.90	**26.73**	27.07
16	29.79	28.39	28.50	28.49

Table 4 Average word error rate (%) for test set 1, combining all models with the EM algorithm based on per utterance perplexity

$\|T\|$	Word model	$\|C\|$ 258	$\|C\|$ 514	$\|C\|$ 1026
1	27.67	27.28	27.07	27.43
4	27.11	26.69	26.61	26.79
8	26.97	26.78	**26.52**	26.77
16	29.20	27.71	28.35	29.00

Table 5 Average word error rate (%) for test set 1, combining all models with the EM algorithm based on per presentation perplexity

$\|T\|$	Word model	$\|C\|$ 258	$\|C\|$ 514	$\|C\|$ 1026
1	27.67	26.86	27.08	27.18
4	26.69	26.37	26.61	26.47
8	26.76	26.28	**26.09**	26.10
16	27.61	26.91	26.90	27.20

shown in Table 3. Table 4 gives results where the EM algorithm is used to adjust the interpolation weights of all models in order to minimize the perplexity on a single utterance. Results obtained when the interpolation weights are adjusted based on the transcription hypothesis of the entire presentation are given in Table 5.

In Tables 3, 4 and 5 the best results are obtained with $|T|$ equal 8 and $|C|$ equal 514.

The next step taken was to experiment on the optimum amount of transcription data used to adjust the weight distribution. Table 6 shows the word error rate when calculating the weight distribution based on hypothesis segments of increasing length. While we initially expected to find a maximum between both extremes, the results show that for presentations of the length found in our development set it is more appropriate to use the entire transcription hypothesis to calculate interpolation weights.

We then investigated in [8] a method proposed by Yokoyama et al. [14], [15] where several combinations of

Table 6 Average perplexity and word error rate (%) for test set 1, combining all models with the EM algorithm based on the perplexity of various window lengths

Adaptation window	$\|T\| = 8$ $\|C\| = 514$	
	Perplexity	Word error rate
Single utterance	93.63	26.52
1/8 of presentation	68.72	26.27
1/4 of presentation	68.37	26.18
1/2 of presentation	68.31	26.10
Whole presentation	68.31	**26.09**

$(D|N|W)$ parameter settings are studied. Table 7 shows the results of experiments where, based on [14], [15], the interpolation weight of $\lambda_1 = 0.30$ and the number of word-classes $|C|$ equal 130 are used. The only variations between the 4 experiments are the settings of the parameter tuple. The first line in the table corresponds to the experiment performed by Yokoyama et al. with the $(T_0|H|H)$ parameter setting. Best results are obtained when both the word-class definition and word-class n-gram model are trained on the entire training corpus and the unigram distribution is taken from the transcription hypothesis with parameter setting $(T_0|T_0|H)$. Another experiment, to build a unigram and perform the subsequent recognition on only half of the transcription hypothesis with the parameter setting $(T_0|T_0|\{\frac{1}{2}\}H)$, led to less than optimal results. This suggests that the amount of training data was not sufficient to create an appropriate model.

Table 7 Average word error rate (%) for test set 1 using word-class trigram with the interpolation weight ($\lambda_1 = 0.30$) and number of word-classes $|C| = 130$ common for each test

Model	Word error rate
G-LM + WC 3-gram LM $(T_0\|H\|H)$	25.22
G-LM + WC 3-gram LM $(T_0\|H\|T_0)$	27.14
G-LM + WC 3-gram LM $(T_0\|T_0\|\{\frac{1}{2}\}H)$	26.61
G-LM + WC 3-gram LM $(T_0\|T_0\|H)$	**24.89**

A desirable improvement to the methods shown in Table 7 is to be able to compute the interpolation weights λ_i in an unsupervised manner.

The intuitive notion that the EM algorithm can not be used to adjust the interpolation weight between the general language model and a word-class model with $(T_0|H|H)$ parameter setting is validated by the distribution weights shown in Table 8. However, by using the $(T_0|T_0|T_0)$ parameter setting to adjust the weight distribution gives results in line with our expectations.

Table 8 Per presentation weight distribution using the EM algorithm depending on the word-class parameter settings used

Presentation	$(T_0\|H\|H)$		$(T_0\|T_0\|T_0)$	
	λ_0	λ_1	λ_0	λ_1
1	0.05	0.95	0.87	0.13
2	0.08	0.92	0.82	0.18
3	0.03	0.97	0.83	0.17
4	0.04	0.96	0.84	0.16
5	0.04	0.96	0.84	0.16
6	0.15	0.85	0.91	0.09
7	0.08	0.92	0.88	0.12
8	0.03	0.97	0.76	0.24
9	0.07	0.93	0.87	0.13
10	0.03	0.97	0.84	0.16
Average	0.06	0.94	0.85	0.15

Table 9 Average word error rate (%) for test set 1 using word-class trigram with the following parameter settings $(T_0|T_0|H)$ $(\lambda_1 = f(EM))$

Model	Word error rate
G-LM + WC 3-gram LM	24.84

After the weight distribution is computed, the unigram component is replaced by the one built on the transcription hypothesis such that the $(T_0|T_0|H)$ parameter settings are used for recognition.

While the average of $\lambda = 0.15$ obtained with the EM algorithm suggests that the results will be sub-optimal according to [14], [15], Table 9 shows that this is not the case. To be able to determine the interpolation weight in an unsupervised manner is important since it allows the method to be applied to different tasks without having to perform empirical calibration on held-out data. This method can also be extended to adjust the weights between more than two models by using the $(T_j|T_j|T_j)$ parameter settings for each word-class model. Since this method is applied in an unsupervised manner, its performance on all test sets will be given in the next section.

4.2 Results on all test sets

Table 10 gives the baseline word error rate and perplexity for the 3 test sets using the general language model. The relative improvement in word error rate compared to the baseline will be given in each of the following tables while perplexity values will not be given for word-class based methods since the unigram component is built from the transcription hypothesis.

In Table 11, results are given for recognition experiments performed with word-class models using

$(T_j|T_j|T_j)$ parameter settings with the weight distribution determined by using the EM algorithm. It should be noted that even though there is a similar reduction in perplexity on test set 3, the reduction in word error rate is much smaller than for other test sets.

Table 12 shows the word error rates obtained by directly using the method described in [14], [15], using the empirically obtained fixed interpolation weight. Table 13 shows the results obtained by using $(T_0|T_0|H)$ parameter settings and estimating the weight of λ_1 with the EM algorithm.

Finally, Table 14 gives the word error rate obtained with our proposed method where $|T| = 8$ clusters are combined with the general language model and word-class models with $(T_0|T_j|H)$ parameter settings are used. The interpolation weight is automatically determined by using the EM algorithm.

Table 10 Average word error rate (%) and perplexity baseline on all test sets using the general language model

Test set	Baseline	
	WER	Perplexity
1 (dev)	27.67	72.33
2	27.05	71.67
3	25.78	90.86

Table 11 Average word error rate (%), perplexity and relative improvement on all test sets using word-class trigrams with the following parameter settings $(T_j|T_j|T_j)$ $|T| = 8$ $|C| = 514$ $(\lambda_j = f(EM))$

Test set	G-LM + WC 3-gram LM		Relative improvement	
	WER	Perplexity	WER	Perplexity
1 (dev)	26.09	68.31	5.71%	5.57%
2	25.65	67.34	5.18%	6.05%
3	25.29	85.65	1.90%	5.73%

Table 12 Average word error rate (%) and relative improvement on all test sets using a word-class trigram with the following parameter settings $(T_0|H|H)$ $|T| = 1$ $|C| = 130$ $(\lambda_1 = 0.30)$

Test set	G-LM + WC 3-gram LM	
	Word error rate	Relative improvement
1 (dev)	25.22	8.85%
2	24.81	8.28%
3	24.48	5.04%

Table 13 Average word error rate (%) and relative improvement on all test sets using a word-class trigram with the following parameter settings $(T_0|T_0|H)$ $|T| = 1$ $|C| = 130$ $(\lambda_1 = f(EM))$

Test set	G-LM + WC 3-gram LM	
	Word error rate	Relative improvement
1 (dev)	24.84	10.23%
2	24.81	8.28%
3	24.75	4.00%

Table 14 Average word error rate (%) and relative improvement on all test sets using word-class trigrams with the following parameter settings $(T_0|T_j|H)$ $|T| = 8$ $|C| = 130$ $(\lambda_j = f(EM))$

Test set	G-LM + WC 3-gram LM	
	Word error rate	Relative improvement
1 (dev)	24.79	10.41%
2	24.25	10.35%
3	24.50	4.97%

5. Discussion

A topic that is particularly relevant to language model adaptation using the transcription hypothesis as a source of information is the length of this hypothesis. As was shown in Table 6 the word error rate gradually decreases as more of the hypothesis output is used to adjust the weights between models. Also, in Table 7, the $p(w|C)$ component of the word-class model built using only half of the output hypothesis performs significantly worse than the one trained on the whole output. In both cases, a smaller amount of adaptation data leads to an increase of the word error rate.

For test set 1, the average length of the output hypothesis is 3176[1] words which means that when using only half of the presentation to adjust the weight distribution, less than 1600 words are available for the adaptation. It is suspected that the relatively small improvement obtained on test set 3 is caused by the short length of the presentations which on average contain only 2167 words, that is about one third less than presentations in test sets 1 and 2.

6. Conclusion

This paper described methods by which it is possible to make further use of readily available information in order to adapt, in an unsupervised way, statistical language models to spontaneous speech recognition tasks.

(1) : While the reference transcription of test set 1 contain 26515 words, the transcription hypothesis contains 31764, including 2126 silence and 3164 short pause tokens.

Among the studied methods, our proposed method of combining a general word n-gram model with word-class n-gram models using the $(T_0|T_j|H)$ parameter setting has given the best relative word error rate reduction of 10.41%, 10.35% and 4.97% on the three CSJ test sets that we have used. While our experiments have shown that the lowest word error rate is obtained when using the entire presentation transcription, this might not always be true for very long presentations. Thus, in future work, it would be important to devise a way of determining how much adaptation data from the transcription hypothesis gives an optimal word error rate reduction. Also, as in the case of test set 3 where the size of the presentation is smaller, it could be useful to extract even more information from the decoder's lattice or to obtain an n-best list of transcriptions in order to increase the amount of adaptation data.

7. Acknowledgments

The authors would like to thank Koji Iwano, Takahiro Shinozaki and Tadasuke Yokoyama for useful discussions that contributed substantially to the work presented in this paper; Takahiro Shinozaki also provided the acoustic models used in our experiments.

References

[1] Philip R. Clarkson and Ronald Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. In *Proceedings EUROSPEECH*, volume 5, pages 2707–2710, Greece, September 1997.

[2] Joshua T. Goodman. A bit of progress in language modeling. Technical report, Microsoft Research, 2001.

[3] Rukmini Iyer and Mari Ostendorf. Modelling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *Proceedings ICSLP*, volume 1, pages 236–239, 1996.

[4] Tatsuya Kawahara, Hiroaki Nanjo, Takahiro Shinozaki, and Sadaoki Furui. Benchmark test for speech recognition using the corpus of spontaneous japanese. In *Proceedings SSPR*, pages 135–138, Tokyo, Japan, 2003.

[5] Reinhard Kneser and Hermann Ney. Improved clustering techniques for class-based statistical language modelling. In *Proceedings EUROSPEECH*, pages 973–976, 1993.

[6] A. Lee, T. Kawahara, and K. Shikano. Julius — an open source real-time large vocabulary recognition engine. In *Proceedings EUROSPEECH*, volume 3, pages 1691–1694, Aalborg, Denmark, 2001.

[7] Luc Lussier, Edward W. D. Whittaker, and Sadaoki Furui. Looking at alternatives within the framework of n-gram based language modeling for spontaneous speech recognition. IEICE SP2003-141, pages 169–174, December 2003.

[8] Luc Lussier, Edward W. D. Whittaker, and Sadaoki Furui. Word-class models for unsupervised language model adaptation applied to spontaneous speech recognition. Acoustical Society of Japan, Spring Meeting, March 2004. To be published.

[9] K. Maekawa, H. Koiso, Sadaoki Furui, and H. Isahara. Spontaneous speech corpus of Japanese. In *Proceedings of LREC*, volume 2, pages 947–952, Athens, Greece, 2000.

[10] Gareth Moore and Steve Young. Class-based language model adaptation using mixtures of word-class weights. In *Proceedings ICSLP*, volume 4, pages 512–515, 2000.

[11] S. Sekine. Automatic sublanguage identification for a new text. In *Second Annual Workshop on Very Large Corpora*, pages 109–120, Kyoto, Japan, 1994.

[12] Takahiro Shinozaki and Sadaoki Furui. Analysis on individual differences in automatic transcription of spontaneous presentations. In *Proceedings ICASSP*, volume 1, pages 729–732, 2002.

[13] Kiyotaka Uchimoto, Chikashi Nobata, Atsushi Yamada, Satoshi Sekine, and Hitoshi Isahara. Morphological analysis of the corpus of spontaneous japanese. In *Proceedings SSPR*, pages 159–162, Tokyo, Japan, 2003.

[14] Tadasuke Yokoyama, Takahiro Shinozaki, Koji Iwano, and Sadaoki Furui. Unsupervised class-based language model adaptation for spontaneous speech recognition. In *Proceedings ICASSP*, volume 1, pages 236–239, Hong Kong, China, 2003.

[15] Tadasuke Yokoyama, Takahiro Shinozaki, Koji Iwano, and Sadaoki Furui. Unsupervised language model adaptation using word classes for spontaneous speech recognition. In *Proceedings SSPR*, pages 71–74, Tokyo, Japan, 2003.