Looking at alternatives within the framework of n-gram based language modeling for spontaneous speech recognition

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
E-mail: {llussier, edw, furui}@furui.cs.titech.ac.jp

Abstract This paper presents different methods using a weighted mixture of word and word-class language models in order to perform language model adaptation. A general language model is built from the whole training corpus, then several numbers of clusters are created according to a word co-occurrence measure and finally, word models as well as word-class models are built from each cluster. The general language model is then combined with one or several other models chosen according to a minimum perplexity criterion. Results show an absolute reduction of the word error rate of 1.40% and 0.49% on average for two different test sets of the “Corpus of Spontaneous Japanese.”

Keyword Spontaneous speech recognition, language model adaptation, document clustering, word models, word-class models, EM algorithm

1. Introduction

While state-of-the-art automatic speech recognition systems can achieve high word accuracy in tasks involving read speech, their performance is not as satisfactory in the case of spontaneous speech recognition tasks. There are several factors that affect spontaneous speech recognition word accuracy, the most significant ones being the speaking rate, out of vocabulary rate and repair rate according to [3].

Thus, for the “Corpus of Spontaneous Japanese” (CSJ)[1], the baseline mean word error rate is slightly under 30%[2]. A plausible explanation for this relatively low word accuracy is the limited amount of training data available for the language modeling of spontaneous speech versus broadcast news speech for example. In order to work around this limitation, we can try to extract more information from our training data by performing language model adaptation.

2. Language model adaptation

2.1. Clustering presentations

All the presentations in the CSJ training corpus were clustered according to the bottom-up method used in [4] and [5], based on [6]. Each presentation \( P \) initially represents a one presentation cluster in a large dimensional space. For each pair of presentations \( P_i \) and \( P_j \), the similarity metric \( S_{ij} \) is defined as

\[
S_{ij} = \sum_{w \in P_i \cap P_j} \frac{N_w}{|P_i| \times |P_j|} \tag{1}
\]
where \(|P^0|\) is the number of presentation clusters that contain the
word \(w\), \(|P_j|\) is the number of unique words in the cluster and \(N_{ij}\) is a
normalization factor used to prevent the development of a single
large cluster

\[
N_{ij} = \frac{N_i + N_j}{N_i \times N_j}
\]  

(2)

with parameters \(N_i\) and \(N_j\) representing the number of presentations
inside each cluster. All words from each presentations where used
to perform the clustering.

2.2. Building word and word-class models

To build new word and word-class models, the 2690
presentations in the training corpus are automatically clustered into
\([T]\) presentation clusters. For each presentation cluster, a word
model is created and the two-sided word clustering algorithm \([7]\) is
used to create \([C]\) word-classes.

2.3. Selecting and weighting models

A general word model is made from all 2690 presentations and
used as a baseline model. This general language model will be
adapted by combining it with models from each combination of \([T]\)
presentation clusters and \([C]\) word-classes.

Below, we look at several ways of selecting and weighting each
model in order to adapt the general language model.

2.3.1. Method I

In Method I, adaptation of the general model is performed on an
utterance-by-utterance basis. There are two versions of this method,
based on the following equation

\[
p(w_i \mid w_{i-1}, w_{i-2}) = \theta_0^C \cdot p_g(w_i \mid w_{i-1}, w_{i-2}) + \sum_{j=1}^{[C]} \theta_j^C \cdot p_s(w_i \mid w_{i-1}, w_{i-2}, T_j)
\]  

(3)

where \(\theta_j\) represents the weight of each model and the sum of all \(\theta_j\)
is equal to 1. \(T_j\) identifies a presentation cluster, \(p_g\) is the general
word language model created from all the available training data and
\(p_s\) is a specialized language model corresponding to either a
presentation-cluster dependent word model

\[
p_s(w_i \mid w_{i-1}, w_{i-2}, T_j) = p(w_i \mid w_{i-1}, w_{i-2}, T_j)
\]  

(4)
or class model

\[
p_s(w_i \mid w_{i-1}, w_{i-2}, T_j) = p(w_i \mid C(w_i), T_j),
\]

\[
p(C(w_i) \mid C(w_{i-1}), C(w_{i-2}), T_j)
\]  

(5)

In the first version, the general language model is combined with
a fixed ratio of 0.65:0.35 with the single model within a group of
models which gives the lowest perplexity for a single utterance.
Thus \(\theta_0\) is equal to 0.65 and the \(\theta_j\) of the chosen model equal
0.35. This ratio, used in \([4]\), was also validated empirically on the
best performing model group.

The second version combines the general model with all the
models for a given \([T]\) and \([C]\), assigning the weight \(\theta_j\) for each
model using the EM algorithm.

This method requires the following steps for each utterance:

- Using the reference language model, perform a first
  recognition of a single utterance;
- Version I: Pick the model, for a given \([T]\) and \([C]\), that
  combined with the general language model gives the lowest
  perplexity on the first-pass transcription;
- Version II: Using the EM algorithm, find the weight
distribution for all models, for a given \([T]\) and \([C]\), as well as the
general language model that minimize the perplexity on the
first-pass transcription;
- Perform a second recognition of the utterance using the
  adapted model.

When combining the general word model only with other word
models and assigning the weights with the EM algorithm, this
corresponds to the mixture-based language model presented in \([9]\).

2.3.2. Method II

This method, also based on equation (3), uses the EM algorithm
to assign weights \(\theta_j\) to all models, for a given \([T]\) and \([C]\), and
involve the following steps for each presentation:

- Using the reference language model, perform a first
  recognition of the whole presentation;
- Using the EM algorithm, find the weight distribution for all
  models, for a given \([T]\) and \([C]\), as well as the general language
  model that minimizes the perplexity of the first-pass transcription
  of the whole presentation;
- Perform a second recognition of the whole presentation using
  the adapted model.

2.3.3. Method III

For comparison purposes, we also tested the method described in
\([4]\) that calculates the word probability of specialized models
according to this equation

\[
p_s(w_i \mid w_{i-1}, w_{i-2}, T_j) = p(w_i \mid C(w_i), T_j),
\]

\[
p(C(w_i) \mid C(w_{i-1}), C(w_{i-2}))
\]  

(6)

instead of using equation (4) and (5). A single class N-gram model
is created from the whole training corpus along with a word
unigram for each presentation cluster.

Method III is tested in the same way as Method II, using the EM
algorithm to assign the weight distribution for all specialized
models as well as to the general language model in order to
minimize the perplexity of the first-pass transcription of the whole
presentation.
3. Experimental conditions

3.1. Acoustic model

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

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 talks are used for the third test set.

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

<table>
<thead>
<tr>
<th>Model</th>
<th># talks (# hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Female</td>
</tr>
<tr>
<td>Academic only</td>
<td>166 (42)</td>
</tr>
<tr>
<td>Academic and extemporaneous</td>
<td>988 (176)</td>
</tr>
</tbody>
</table>

3.2. Baseline language model

The training corpus consists of 2690 presentations, providing almost 7.5 million words with a vocabulary size of 30678 words. As in [3], the term “word” refers to the Japanese morphemes. All of the training data is used to build a forward word bi-gram and a reverse word tri-gram as needed by Julius. This model is referred to as the “general language model” in this paper. A variation of the smoothing technique developed by Kneser and Ney introduced in [8] is used with all language models.

3.3. Speech recognition engine

The recognition is performed with the Julius version 3.3p3 engine. We have slightly modified Julius in order to obtain language model probabilities from an external library. This allowed us to experiment with the combinations of several language models, something that was not previously possible with Julius.

4. Experimental results

We use the first of the three test sets defined in the CSJ benchmark [1] as a development set and use the last two for testing. Test set one and two contain academic presentations and test set three extemporaneous presentations. Each test set contains 10 presentations.

4.1. Development set results

4.1.1. Perplexity

Perplexity values for all methods display a similar tendency to decrease noticeably as the number of presentation clusters increase. Values also tend to increase slightly as the number of word-classes increase.

Method I gives higher perplexity values than Methods II and III since it represents the average perplexity per utterance rather than the average perplexity per presentation and thus these values should not be directly compared.

Tables 2 and 3 show perplexity values for the two versions of Method I. We notice that using the EM algorithm to adjust the language model weights gives a greater reduction in perplexity compared to using only the single-best model.

Table 2: Method I (Version I), average sentence perplexity for the combination of the general and one best model.

<table>
<thead>
<tr>
<th>[T]</th>
<th>Word model</th>
<th>258</th>
<th>514</th>
<th>1026</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114.79</td>
<td>105.72</td>
<td>106.54</td>
<td>110.73</td>
</tr>
<tr>
<td>4</td>
<td>105.19</td>
<td>101.46</td>
<td>102.83</td>
<td>105.36</td>
</tr>
<tr>
<td>8</td>
<td>99.47</td>
<td>98.52</td>
<td>100.67</td>
<td>101.10</td>
</tr>
<tr>
<td>16</td>
<td>95.85</td>
<td>96.21</td>
<td>96.80</td>
<td>96.38</td>
</tr>
</tbody>
</table>

Table 3: Method I (Version II), average sentence perplexity, EM weighted combination of all models.

<table>
<thead>
<tr>
<th>[T]</th>
<th>Word model</th>
<th>258</th>
<th>514</th>
<th>1026</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114.79</td>
<td>103.79</td>
<td>105.12</td>
<td>109.77</td>
</tr>
<tr>
<td>4</td>
<td>95.87</td>
<td>98.44</td>
<td>99.11</td>
<td>101.46</td>
</tr>
<tr>
<td>8</td>
<td>85.67</td>
<td>92.21</td>
<td>93.63</td>
<td>93.55</td>
</tr>
<tr>
<td>16</td>
<td>78.00</td>
<td>86.39</td>
<td>84.78</td>
<td>83.66</td>
</tr>
</tbody>
</table>

Looking at the average presentation perplexity results, Tables 4 and 5 also show some improvement compared to the general word language model. It is interesting to notice that while the perplexity decreases by at least one point with Method II each time the number of clusters increase, it varies much less with Method III. This is probably due to the class N-gram component remaining constant and the contribution from the unigram component having only a small variation regardless of the number of clusters used in Method III. In Method II on the other hand, both the unigram component and the class N-gram component are different for each presentation cluster.

Table 4: Method II, Average presentation perplexity, EM weighted combination of all models.

<table>
<thead>
<tr>
<th>[T]</th>
<th>Word model</th>
<th>258</th>
<th>514</th>
<th>1026</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72.33</td>
<td>70.62</td>
<td>70.98</td>
<td>71.37</td>
</tr>
<tr>
<td>4</td>
<td>68.08</td>
<td>69.19</td>
<td>69.90</td>
<td>69.77</td>
</tr>
<tr>
<td>8</td>
<td>66.18</td>
<td>68.30</td>
<td>68.31</td>
<td>68.20</td>
</tr>
<tr>
<td>16</td>
<td>65.21</td>
<td>67.52</td>
<td>67.28</td>
<td>66.93</td>
</tr>
</tbody>
</table>
4.1.2. Word error rate

While a reduction in perplexity is often an indicator of a reduction in word error rate, we observe the opposite effect when using 16 clusters with all models. This is likely caused by the small number of words per cluster, 467 thousand on average with 16 clusters, over-fitting the adapted model to a partially correct hypothesis.

In accordance with the results obtained for perplexity with Method I, Tables 6 and 7 show slightly lower word error rate is obtained when adjusting the language model weights with EM rather than using the single-best model.

### Table 6: Method I (Version I), average word error rate (%) for the combination of the general and one best model.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C = 514</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>27.67</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>27.15</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>26.92</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>29.79</td>
</tr>
</tbody>
</table>

### Table 7: Method I (Version II), average word error rate (%) for the EM weighted combination of all models.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C = 514</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>27.67</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>27.11</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>26.97</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>29.20</td>
</tr>
</tbody>
</table>

4.2. Test set results

We present the results from Method II that provided the greatest improvements in our development set with |T| = 8 clusters and |C| = 514 word-classes. While our development set consists of only male speakers, the second and third CSJ test sets contain speech by both female and male speakers. The context of the first two test sets is academic presentations, and extemporaneous presentations for the third one.

4.2.1. Perplexity

As shown in Table 10, even if the perplexity is higher in the case of extemporaneous presentations, for all test sets Method II gives a reduction in perplexity of 4.5 points on average.

### Table 10: Comparison of average presentation perplexity values obtained for each test set between the reference model and Method II with 8 presentation clusters and 514 word-classes.

<table>
<thead>
<tr>
<th>Test set</th>
<th>Reference</th>
<th>Method II</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (dev)</td>
<td>72.33</td>
<td>68.31</td>
</tr>
<tr>
<td>2</td>
<td>71.67</td>
<td>67.34</td>
</tr>
<tr>
<td>3</td>
<td>90.86</td>
<td>85.65</td>
</tr>
</tbody>
</table>

4.2.2. Word error rate

With a comparable reduction in perplexity for each test set, Table 11 shows that the reduction in word error rate is much smaller in the case of the extemporaneous presentations. It is not clear why the improvement is smaller in the case of test set number 3. A possible explanation is that there is 1.5 times more data available to model extemporaneous speech than academic presentations thus resulting in a smaller gain in word error rate.

### Table 8: Method II, average word error rate (%) for the EM weighted combination of all models using 514 word-classes.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C = 514</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>27.08</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>26.70</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>26.59</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>27.25</td>
</tr>
</tbody>
</table>

### Table 9: Method III, average word error rate (%) for the EM weighted combination of all models using 514 word-classes.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C = 514</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>26.70</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>26.59</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>27.25</td>
</tr>
</tbody>
</table>
reducing the effect of the adaptation.

Table 11: Comparison of word error rate (%) values obtained for each test set between the reference model and Method II with 8 presentation clusters and 514 word-classes.

<table>
<thead>
<tr>
<th>Test set</th>
<th>Reference</th>
<th>Method II</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (dev)</td>
<td>27.67</td>
<td>26.09</td>
</tr>
<tr>
<td>2</td>
<td>27.05</td>
<td>25.65</td>
</tr>
<tr>
<td>3</td>
<td>25.78</td>
<td>25.29</td>
</tr>
</tbody>
</table>

Nevertheless, to verify that the assignment of weights reflect the content of each test set, we have plotted in figure 1 the average weight given to each cluster per test set. As expected, the distribution for test sets 1 and 2 is similar and clusters receiving a near zero weight for academic presentations are given more weight for extemporaneous presentations and vice versa.

Figure 1: Language model weight distribution per test set.

5. Discussion

From the different methods that we have compared in this paper, we observe that compared with the general language model, adding a mixture of word models can lower the word error rate. Moreover, a mixture of word class models can even further help to reduce the word error rate providing that we do not divide the training data so much as to end up with poorly trained models.

The amount of data from the first-pass transcription used to adjust the weights for the model adaptation also has some effect. Method I and II illustrate both extremes of this tradeoff by using a single utterance in the former method and the whole presentation in the latter. While our current results show that using the whole presentation gives better results, we intend to further investigate the effect of using different amount of first-pass transcription for adjusting the model weights with the aim of obtaining more refined adaptation.

While Method III performed slightly worse than Method II, there is a tradeoff between the performance and the amount of space used by the language model. However, our results do show that using more word class models can improve the accuracy.

6. Summary and conclusion

We have presented a new method of adapting language models for spontaneous speech recognition that gave an absolute reduction of 1.40% and 0.49% in word error rate on the second and third test sets of the CSJ corpus.

Further research will investigate different ways of selecting the presentations used during the adaptation process and also look at other criteria for clustering the selected presentations. It will also be of interest to look for ways to model the speaker style separately from the presentation’s topic, especially when the amount of training data is still limited.

References