Combinations of Various Language Model Technologies including Data Expansion and Adaptation in Spontaneous Speech Recognition

Ryo Masumura†, Taichi Asami†, Takanobu Oba†, Hirokazu Masataki†, Sumitaka Sakauchi†, Akinori Ito‡
† NTT Media Intelligence Laboratories, NTT Corporation, Japan
‡ Graduate School of Engineering, Tohoku University, Japan
†{masumura.ryo, asami.taichi, oba.takanobu, masataki.hirokazu, sakauchi.sumitaka}@lab.ntt.co.jp
‡aito@spcom.ecei.tohoku.ac.jp

Abstract
This paper demonstrates combinations of various language model (LM) technologies simultaneously, not only modeling techniques but also those for training data expansion based on external language resources and unsupervised adaptation for spontaneous speech recognition. Although forming combinations of various LM technologies has been examined, previous works focused on only modeling techniques. In fact, the previous works did not consider other important functionalities in practical spontaneous language modeling; a use of external language resources and an unsupervised LM adaptation. Therefore, our examination employs not only manual transcriptions of target domain speech but also out-of-domain text resources for spontaneous language modeling. In addition, the unsupervised LM adaptation based on multi-pass decoding is aggressively introduced to the combination. Our experimental results show a significant word error rate reduction by combining various technologies compared to using each technology individually in Japanese spontaneous speech recognition task. Furthermore, we also reveal relationships between the technologies.

Index Terms: combination of various language model technologies, external language resources, unsupervised adaptation, spontaneous language modeling

1. Introduction
Recent speech recognition technologies have accomplished remarkable progress in terms of practical realization [1]. They are now used in various applications such as dictation and voice search, and are attracting wide attention. Modern practical speech recognition systems employ two types of statistical models: acoustic and language models (LMs). This framework was established many decades ago [2]. It can be said that current progress in speech recognition technology is realized as advancements in these two types.

In recent years, a major breakthrough occurred in acoustic modeling with the introduction of the deep neural network (DNN) [3]. DNNs catch acoustic features more precisely than traditional models, and DNN-based acoustic modeling has achieved significant improvements [4]. With regard to language modeling, however, there have been no comparable breakthroughs for a long time. It is clear that modern practical LMs, i.e. word n-gram models, have several problems [5]. Although various techniques for tackling these problems have been proposed, including DNN-based language modeling [6], the performance improvements offered by introducing individual techniques remain insufficient.

In consideration of this fact, combining various LM technologies has been examined [7, 8, 9]. Previous works showed that this approach may yield much better performance than the single use of any one of them. This knowledge is very useful for language modeling; however, studies published to date merely considered the modeling techniques and failed to conduct examinations involving practical spontaneous speech recognition systems.

In fact, two important issues are raised in making a practical spontaneous speech recognition system. The first one is the scarcity of training data. For instance, in the voice search task, it is easy to collect the data corresponding to the target task by accessing query logs [10]. In the spontaneous speech task, on the other hand, the data corresponding to the target task must be obtained by manually transcribing speech. Thus, data expansion techniques that can collect useful training data sets from external text resources are important [11, 12, 13]. In addition, data expansion techniques are useful in addressing the out-of-vocabulary (OOV) problem. The second issue is the current weakness of unsupervised adaptation [14]. Since technologies that robustly perform in various tasks are difficult to realize, it is important to make LMs that specialize in processing input speech. Unsupervised adaptation is the most attractive approach for realizing these kinds of technologies.

Therefore, in this paper, we try to use various LM technologies simultaneously, not only modeling techniques but also data expansion and unsupervised adaptation. To this end, we divide LM technologies into several categories and investigate the effectiveness of combining multiple LM technologies categorically and all of them in Japanese spontaneous speech recognition task. Furthermore, our investigation also aims to reveal the interaction between the techniques.

This paper is organized as follows. Section 2 describes our categorization of LM technologies. Details of LM technologies for each category and how to combine multiple techniques are described in Sections 3. Section 4 describes our experiments and discusses the relationship between the techniques. Section 5 concludes this paper.

2. Categorization of Language Model Technologies

2.1. Baseline Language Model
Given manual transcriptions of a target spontaneous speech task, our baseline system consists of back-off n-gram models trained from them [15]. The back-off n-gram modeling method...
is widely used because of its compactness, powerfulness, and suitability for the weighted finite state transducer (WFST) based decoding technology [16, 17]. This paper uses a hierarchical Pitman-Yor LM (HPYLM) as the n-gram model [18]. HPYLM is a theoretically elegant Bayesian n-gram model that has demonstrated top performance among several smoothing methods [19].

2.2. Categorization

This paper introduces different kinds of LM technologies. A consideration of the applicable scope of each technology allows us to divide LM technologies into three categories.

• **Direct decoding**: Direct (one pass) decoding is the ideal form of speech recognition. It basically demands the use of an n-gram model which is suitable for WFST decoders that support on-the-fly composition techniques [17]; we then introduce some techniques to construct robust n-gram models specific to the target task.

• **Unsupervised adaptation**: With WFST-based decoding, we can achieve further improvement through the unsupervised adaptation of an n-gram model. Unsupervised adaptation can capture long-range information of the processing input speech using its recognition hypotheses.

• **Rescoring**: Rescoring is introduced after WFST-based decoding. It allows us to utilize techniques that are difficult to directly introduce to decoding. Rescoring is performed after hypotheses are obtained, so it makes it easy to apply complicated techniques.

3. Various Language Model Technologies

3.1. Techniques for Direct Decoding

3.1.1. HPYLMs as an Approximation of Other Models

To construct robust n-gram models, we can introduce techniques that convert different kinds of LMs into n-gram models [20, 21]. In this framework, a LM is constructed from training data for text generation, and then texts are generated by the model via random sampling. After that, an n-gram model is trained using the generated data.

In this paper, we use latent words LMs (LWLMs) for text generation [20]. LWLMs are generative models, where each latent variable is associated with an observed word [22]. It is expected that the generated data contains various linguistic expressions that are not contained in the original training data. In constructing the HPYLM from generated data, entropy based pruning can be used to reduce the model size [23].

3.1.2. HPYLMs Using External Text Resources

An alternative technique for constructing robust n-gram models is based on data expansion from external text resources [11, 12]. As the data expansion method, this paper uses the difference in entropy between in-domain and out-of-domain models [24]. In this framework, the in-domain LM is constructed from training data, while the out-of-domain LM is constructed from data randomly extracted from external text resources. $H_I(s)$ is the entropy of sentence $s$ within the in-domain LM, $H_O(s)$ is the entropy of $s$ within the out-of-domain LM; the sentence score $D(s)$ is defined as:

$$D(s) = H_I(s) - H_O(s).$$

We collect sentences whose score is less than threshold $T$ as training data to construct the HPYLM. In this paper we set $T$ to 0, which is equivalent to using the Bayes classifier for sentence selection [13].

3.2. Techniques for Unsupervised Adaptation

3.2.1. Topic Model Based Adaptation

In unsupervised adaptation with topic models, the entire topic information of the processing target speech is determined by the recognition hypothesis generated in the first pass, and the n-gram model is adapted using the estimated topic. This paper uses the unigram marginal technique since it allows back-off probabilities in the n-gram model to be considered [25, 26]. We use latent Dirichlet allocation (LDA) as the topic model [27].

In this framework, the topic probability is estimated using a recognition hypothesis of the processing speech, and next the n-gram model is adapted using the adapted unigram probability. Adapted n-gram probability of word $w$ in given context $u$ is given by:

$$P_k(w|u) = \left( \sum_z P(w|z)P(z|d) \right)^\mu P_0(w|u) Z(u),$$

where $z$ is a topic, and $d$ is a recognition hypothesis. $P(w|z)$ and $P(z|d)$ are calculated based on LDA. $\mu$ is a tuning parameter. $P_0(w|u)$ and $P_0(u)$ is the n-gram model used in the first pass. $Z(u)$ is a normalization term.

3.2.2. Document Retrieval Based Adaptation

Document retrieval based unsupervised adaptation can be split into the following steps. First, relevant data is selected from external text resources using a recognition hypothesis of the processing speech. Next, the n-gram model is adapted using relevant documents [28, 29, 30]. In this paper, we use vector-space-model-based document retrieval for document expansion [29].

Document retrieval based on the vector space models uses cosine similarity between the recognition hypothesis and the external text. After retrieval, unsupervised adaptation is conducted by mixing the baseline n-gram model with the n-gram model constructed from the relevant documents. Adapted n-gram probability of word $w$ in given context $u$ is given by:

$$P_k(w|u) = \lambda P_0(w|u) + (1 - \lambda) P_0(w|u),$$

where $P_0(w|u)$ is the n-gram model used in the first pass and $P_0(w|u)$ is the n-gram model constructed from relevant documents. $\lambda$ is a mixture weight that is automatically optimized using the recognition hypothesis [31].

3.3. Techniques for Rescoring

3.3.1. Recurrent Neural Network Language Models

The recurrent neural network LM (RNNLM) has attracted significant attention in recent years [32]. RNNLMs have two characteristics: one is that the word space can be represented as a continuous space vector based on neural networks, and the other is that long-range information can be flexibly taken into consideration based on its recurrent structure. Since using an RNN makes the cost of computing the probability estimation proportional to the lexical size of the output layer, class-based RNNLMs are most commonly used [33]. The resulting probability estimation is defined as:

$$P(w|u) = P_{rnn}(w|s, c)P_{rnn}(c|s),$$
where $s$ is context information, which includes previous word and previous output in hidden layer, and $c$ means word class. $P_{\text{RNN}}(w|s, c)$ and $P_{\text{RNN}}(c|s)$ can be calculated based on trained RNN. In rescoring, we use the probability obtained by linearly interpolating an n-gram model and RNNLM.

3.3.2. Discriminative Language Models

Discriminative LMs (DLMs) can evaluate whether a recognition hypothesis is correct or incorrect [34]. DLMs are also called error corrective models or re-ranking models. The n-best list generated from a speech recognizer is denoted as $L = \{d_j | j = 1, \ldots, m\}$ where $d_j$ is the $j$-th hypothesis in the n-best list. Error correction using a DLM is realized as:

$$d^* = \arg \max_{d \in L} \{a_0 f_0(d) + \alpha^T f(d)\}, \quad (5)$$

where $f(d)$ is the feature vector of $d$ and $f_0(d)$ is the speech recognition score of $d$. $a_0$ and $\alpha$ denote a scaling factor and model parameter of DLM, respectively. There are several methods to estimate the model parameter [35]. We use round-robin dual discrimination (R2D2) for this estimation [36].

3.4. Combination of Multiple Techniques

We combine the multiple language modeling techniques with baseline LM in each category.

In direct decoding, each technique generates an n-gram model as well as a baseline LM, so we can combine the techniques by using the n-gram mixture model approach. The resulting n-gram mixture model can be approximated as a single back-off n-gram model [37], so it can be also used in direct decoding. The mixture weights are preliminarily optimized using a validation set and the EM algorithm.

In unsupervised adaptation, adapted models based on each technique are expressed as n-gram models, so we can also combine the techniques as n-gram mixture models. In contrast to direct decoding, the mixture weights are optimized using the recognition hypothesis generated in the first pass.

In rescoring, RNNLM has a different structure from DLM, so each technique is introduced in series. In fact, DLM must be introduced at the end because it is used for error correction. RNNLM-based rescoring uses a mixed score yielded by n-gram model and RNNLM. After adding RNNLM, DLM-based rescoring is conducted. In this case, the speech recognition score contains the RNNLM-based score. The mixture weight of RNNLM and the scaling factor of DLM are preliminarily defined by the validation set.

4. Experiments

4.1. Experimental Setup

Our experiments used the Corpus of Spontaneous Japanese (CSJ) [38]. CSJ was divided into a training set, validation set, and test set. The validation set was used for optimizing several hyper parameters. Details of the data sets are shown in Table 1.

We also prepared text data of about 50 billion morphemes from the web as the external text resource. We used an acoustic model based on hidden Markov models with deep neural networks (DNN-HMM) [3, 4]. The trained DNN-HMM had 8 hidden layers with 2048 nodes and 3072 outputs. The speech recognition decoder was VoiceRex, a WFST-based decoder [17, 39]. JTAG was used as the morpheme analyzer to split sentences into words [40].

<table>
<thead>
<tr>
<th>Data</th>
<th># of lectures</th>
<th># of morphemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>2,472</td>
<td>6,752,588</td>
</tr>
<tr>
<td>Training for DLM</td>
<td>200</td>
<td>542,215</td>
</tr>
<tr>
<td>Validation</td>
<td>10</td>
<td>28,547</td>
</tr>
<tr>
<td>Test 1</td>
<td>10</td>
<td>28,504</td>
</tr>
<tr>
<td>Test 2</td>
<td>10</td>
<td>18,426</td>
</tr>
</tbody>
</table>

Our evaluation examined the following methods. We combined these methods category-wise and then all of them.

1. **Baseline HPYLM**: 3-gram HPYLM constructed from the training set [19]. No pruning technique was used. Vocabulary size was 78K words.

2. **HPYLM based on LWLM**: 3-gram HPYLM constructed from 1 billion morphemes generated based on LWLM [20]. LWLM was constructed from the training set. Entropy based pruning was conducted. Vocabulary size was 78K words which correspond to the baseline.

3. **HPYLM using external resources**: 3-gram HPYLM constructed from 2 billion morphemes selected from the external text resources [13]. Entropy based pruning was used. Vocabulary consisted of 78K words which correspond to the baseline. A 600K word model was also prepared as vocabulary expansion variant.

4. **Topic model based adaptation**: Unsupervised adaptation using unigram marginal technique and LDA [26]. LDA was constructed from a training set containing 50 topics. Tuning parameter was set to 0.5. The vocabulary size of the adapted model was 78K.

5. **Document retrieval based adaptation**: Unsupervised adaptation based on document retrieval using vector space model [29]. We selected top 1K documents from external text resources and adapted an n-gram model. The vocabulary size of the adapted model was restricted to 78K which corresponds to the baseline. Moreover, a vocabulary-limitation-free model was also prepared as a vocabulary expansion variant.

6. **RNNLM**: RNNLM constructed from the training set [33]. It used 500 hidden neurons, 1000 classes. Vocabulary size was 78K words which correspond to the baseline.

7. **DLM**: DLM with word features constructed from training data for DLM. R2D2 method was used for training [36]. To generate recognition hypotheses about the training data, we used two kinds of models. One is the baseline system and the other is a system that combined all techniques except DLM. The latter is denoted as $7^*$.  

4.2. Experimental Results

Table 2 shows the perplexity (PPL) and word error rate (WER) results for the validation set and each test set. We evaluated both limited vocabulary size and vocabulary expansion variants. Incidentally, the difference of PPL in the vocabulary expansion variants cannot be compared since each variant has different vocabulary size. In addition, there are no PPL results for DLM since DLM is an error corrective model for recognition results. We labeled the experimental conditions from (a) to (c) where each number corresponds to our experimental setup.
Table 2: Experimental results: perplexity and word error rate (%).

<table>
<thead>
<tr>
<th>Category</th>
<th>Setup</th>
<th>Vocabulary expansion</th>
<th>Validation</th>
<th>Test 1</th>
<th>Test 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PPL</td>
<td>WER</td>
<td>PPL</td>
</tr>
<tr>
<td>(a) Baseline</td>
<td>1.</td>
<td></td>
<td>83.18</td>
<td>20.01</td>
<td>70.72</td>
</tr>
<tr>
<td>(b) Direct decoding</td>
<td>1+2.</td>
<td></td>
<td>77.86</td>
<td>19.16</td>
<td>67.42</td>
</tr>
<tr>
<td>(c) Direct decoding</td>
<td>1+3.</td>
<td></td>
<td>77.77</td>
<td>18.84</td>
<td>67.09</td>
</tr>
<tr>
<td>(d) Direct decoding</td>
<td>1+2+3.</td>
<td></td>
<td>74.89</td>
<td>18.45</td>
<td>64.92</td>
</tr>
<tr>
<td>(e) Unsupervised adaptation</td>
<td>1+4.</td>
<td></td>
<td>71.87</td>
<td>19.29</td>
<td>64.87</td>
</tr>
<tr>
<td>(f) Unsupervised adaptation</td>
<td>1+5.</td>
<td></td>
<td>71.47</td>
<td>18.38</td>
<td>64.82</td>
</tr>
<tr>
<td>(g) Unsupervised adaptation</td>
<td>1+4+5.</td>
<td></td>
<td>65.24</td>
<td>18.07</td>
<td>60.60</td>
</tr>
<tr>
<td>(h) Rescoring</td>
<td>1+6.</td>
<td></td>
<td>71.70</td>
<td>19.01</td>
<td>63.29</td>
</tr>
<tr>
<td>(i) Rescoring</td>
<td>1+7.</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(j) Rescoring</td>
<td>1+6+7.</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(k) Whole</td>
<td>1+2+3+4+5.</td>
<td></td>
<td>64.53</td>
<td>17.49</td>
<td>59.89</td>
</tr>
<tr>
<td>(l) Whole</td>
<td>1+2+3+4+5+6.</td>
<td></td>
<td>61.32</td>
<td>17.23</td>
<td>56.74</td>
</tr>
<tr>
<td>(m) Whole</td>
<td>1+2+3+4+5+6+7.</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(n) Whole</td>
<td>1+2+3+4+5+6+7+².</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(o) Direct decoding</td>
<td>1+2+3.</td>
<td>✓</td>
<td>81.12</td>
<td>18.12</td>
<td>68.58</td>
</tr>
<tr>
<td>(p) Unsupervised adaptation</td>
<td>1+4+5.</td>
<td>✓</td>
<td>69.01</td>
<td>17.88</td>
<td>63.97</td>
</tr>
<tr>
<td>(q) Whole</td>
<td>1+2+3+4+5.</td>
<td>✓</td>
<td>69.93</td>
<td>17.26</td>
<td>62.85</td>
</tr>
<tr>
<td>(r) Whole</td>
<td>1+2+3+4+5+6.</td>
<td>✓</td>
<td>67.32</td>
<td>17.08</td>
<td>60.62</td>
</tr>
<tr>
<td>(s) Whole</td>
<td>1+2+3+4+5+6+7.</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(t) Whole</td>
<td>1+2+3+4+5+6+7+².</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

First, we evaluated the effectiveness of combining various LM technologies under the constraint of limited vocabulary size. We could achieve performance improvements by combining individual technologies compared to baseline (a). Moreover, we could obtain further improvements by combining LM technologies in each category compared to using only a single technique in each category. For instance, (d) is more effective than (b) or (c). This result shows that individual technologies in each category can complement each other. The best performance was obtained when all technologies were combined and matched DLM was used. In terms of WER, a 4.1 point improvement in test 1 and a 1.5 point improvement in test 2 were achieved by (n) compared to baseline (a).

Additionally, further WER improvements were achieved by vocabulary expansion. In direct decoding, a 0.7 point improvement in test 1 and a 0.3 point improvement in test 2 were achieved by (g) compared to limited vocabulary variant (k) in terms of WER. The highest performance was attained by combining all of techniques with vocabulary expansion condition (t).

Next, we discuss the relationship between the technologies. Five significant points can be extracted from Table 2.

First, building a robust n-gram model for direct decoding, (d), yielded a 2.5 point WER improvement in test 1 and a 3.7 point improvement in test 2 compared to baseline (a). This result shows that a remarkable improvement is possible by constructing a robust model from voluminous training data even if the model structure is an n-gram model.

Second, there was only a slight improvement in performing unsupervised adaptation after constructing the robust n-gram model. While the difference in WER between (a) and (g) was 2 points in test 1 and 3 points in test 2, the difference between (d) and (k) was 0.5 points in each test set. Even if we build a robust n-gram model, the long-range information is not well reflected. This result shows that using the long-range information offers comparatively small benefit.

Third, the improvement offered by RNNLM was small after performing robust n-gram modeling and unsupervised adaptation. While the difference in WER between (a) and (h) was about 1 point in each test set, the difference between (k) and (l) was about 0.2 points. This shows that n-gram modeling offers effects similar to RNNLM by raising the robustness based on data expansion and reflecting the long-range information based on unsupervised adaptation.

Fourth, DLM always demonstrated a fixed improvement even if used at the end. The WER difference between (a) and (i) was virtually the same as that between (l) and (m). This attributed to the fact that DLMs have a different aspect from other modeling techniques. There are parts which are not solved if only correct word sequences are modeled, so the framework of modeling the speech recognition error directly is an effective solution. However, the performance improvement between (s) to (t) was slight even though DLM was trained using the recognition hypothesis generated from a matched system.

Finally, vocabulary expansion was effective although extra words were also increased. In fact, the OOV rate in (o) was improved from 0.54 % to 0.10 % in test 1, and from 1.08 % to 0.07 % in test 2 compared to (d). It seems that these OOV rate improvements induced the WER improvements.

5. Conclusions

In this paper, we examined the combination of various LM technologies including data expansion via external language resources and unsupervised adaptation in the spontaneous speech recognition task. We demonstrated that significant performance improvements were possible by combining various technologies, compared to using each technology individually. Combining all of technologies does lead to 20-25% relative reduction in WER. Furthermore, our investigation revealed several remarkable facts: the power of robust n-gram modeling, the relationship between RNNLM rescoring or unsupervised adaptation and other technologies, the uniqueness of DLM, and the importance of vocabulary expansion.
6. References


