LANGUAGE MODEL DOMAIN ADAPTATION VIA RECURRENT NEURAL NETWORKS WITH DOMAIN-SHARED AND DOMAIN-SPECIFIC REPRESENTATIONS

Tsuyoshi Morioka[1], *Naohiro Tawara*[1], *Tetsuji Ogawa*[1]
Atsunori Ogawa[2], *Tomoharu Iwata*[2], *Tetsunori Kobayashi*[1]

[1] Department of Computer Science, Waseda University, Tokyo, Japan
[2] NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

ABSTRACT

Training recurrent neural network language models (RNNLMs) requires a large amount of data, which is difficult to collect for specific domains such as multiparty conversations. Data augmentation using external resources and model adaptation, which adjusts a model trained on a large amount of data to a target domain, have been proposed for low-resource language modeling. While there are the commonalities and discrepancies between the source and target domains in terms of the statistics of words and their contexts, these methods for domain adaptation make the commonalities and discrepancies jumbled. We propose novel domain adaptation techniques for RNNLM by introducing domain-shared and domain-specific word embedding and contextual features. This explicit modeling of the commonalities and discrepancies would improve the language modeling performance. Experimental comparisons using multiparty conversation data as the target domain augmented by lecture data from the source domain demonstrate that the proposed domain adaptation method exhibits improvements in the perplexity and word error rate over the long short-term memory based language model (LSTMLM) trained using the source and target domain data.

Index Terms— recurrent neural network, language models, data augmentation, domain adaptation

1. INTRODUCTION

Neural network based language models (NNLMs) deal with contextual information using compact continuous vector representations, and yield significant improvements over n-gram language models. In particular, recurrent neural network based language models (RNNLMs) [1, 2], which have become a standard technology for automatic speech recognition (ASR), can deal with an arbitrary length of contexts by unfolding time-delay feedback connections through time.

In general, training RNNLMs requires a large amount of data, which is difficult to collect for specific domains such as multiparty conversations. Several attempts have been made to develop robust RNNLMs for the case when large-scale data are not available during training. The simplest way to address this issue is to augment the training data using external resources [3, 4]. In addition, domain adaptation techniques [5, 6, 7], such as fine-tuning network weights and inserting adaptation layers into the network, have been proposed. These methods adjust a language model that is trained on a large amount of data to a specific target domain. However, they do not explicitly consider the commonality and discrepancy in the statistics of words and their contexts between the source and target domain. For example, transcriptions of multiparty conversations contain phenomena specific to the multiparty conversations as well as those observed also in other domains such as lecture talks. Specifically, back-channel feedbacks frequently appear in multiparty conversations, but not in lecture talks. On the other hand, a speaker who takes the initiative in a conversation conveys a substantial amount of information while speaking. In this case, the spoken words and their context have characteristics similar to those that appear in lecture talks. It should be noted that phenomena that are specific to conversations (e.g., back-channel feedbacks) and those that commonly occur in conversations and lectures (e.g., propagation of substantial information) dynamically alternate. As both the common phenomena between the source and target domain and specific phenomena for each domain would affect the prediction of subsequent word respectively, domain-shared and domain-specific statistics related to words and their contexts should be accumulated separately. Depending on the inputs, the domain-shared or domain-specific phenomena handling should be switched adaptively inside the model. On the other hand, implementing data augmentation and model adaptation without considering the commonality and discrepancy between the source and target domains directly leads to obscuring statistics related to words and their contexts.

We propose domain adaptation techniques for RNNLMs, in which domain-shared and domain-specific representations of words and their contexts are incorporated into a RNNLM. Specifically, word embedding and contextual representations are extended into those that consist of domain-shared, source domain-specific and target domain-specific elements. When the inputs belong to the target domain, the RNNLM propagates with the domain-shared and target domain-specific representations activated. The parameters for corresponding elements are estimated without changing the training algorithms. This method, which is inspired by the technique of feature augmentation approach for domain adaptation [8], is advantageous over existing domain adaptation techniques as it does not jumble the domain-shared and domain-specific statistics of words and their contexts.

The rest of the present paper is organized as follows. Section 2 describes relevant previous work on domain adaptation for language models. Section 3 briefly reviews the RNNLMs. Section 4 proposes RNNLMs with domain-shared and domain-specific representations. Section 5 demonstrates the effectiveness of the proposed method using a multiparty conversation dataset as the target domain and a lecture dataset as the source domain. Finally, the summary is presented in Section 6.

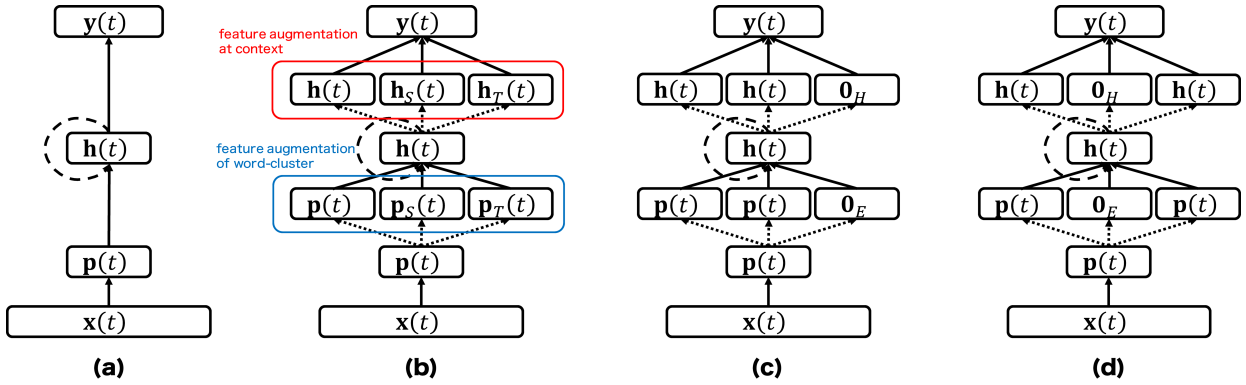

Fig. 1. *Illustration of (a) coventional RNNLM, and (b) RNNLM with domain-shared and domain-specific representations. Dashed line indicates time-delayed feedback connections. Red and blue box depict feature augmentation of context and word occurence information, respectively. Dotted line indicates copying state of hidden layers or padding zero vector. (c) depicts states of network given source domain data and (d) depicts states of network given target domain data.*

2. RELEVANT PREVIOUS WORK

Many attempts have been made to develop robust language models for limited resources. In this section, existing work on data augmentation, model-based adaptation, and feature augmentation, which are useful for RNNLMs, are described.

Data augmentation [3, 4, 9, 10, 11] is a technique to expand the training data. In this, data with characteristics similar to the target domain are chosen from external resources on the basis of the cross entropy between the external and target data to augment the training data [3, 4, 9]. However, in some circumstances, obtaining external resources that are similar to the target domain would be difficult. For example, consider a case with spontaneous speech data and webdata as the target and source domain, respectively. Here, fillers and short-pauses, which are specific to spoken languages, had to be inserted to synthesize spoken languages from the webdata [4]. Conversational data often contain more complicated phenomena such as turn-taking, which are difficult to simulate. An alternate way to augment the training data is by using variational approximation [10, 11], which generates additional words from the RNNLMs. However, there is no guarantee that the data generated by the RNNLMs have a proper meaning. In contrast, the proposed method can handle domain-shared and domain-specific phenomena distinctively, and does not require additional data augmentation to bring data from external resources closer to the target domain (e.g., insertion of fillers to transcriptions from the web).

Model-based adaptation attempts to adjust a source model to the target domain under the assumption that the source model has already been trained on a large amount of external resources. One of the simple solutions is fine-tuning. If the training data contain several domains, the NNLMs are trained on the whole data and then fine-tuned for a specific domain [6]. Linear input networks (LINs) [5], which insert an adaptation layer between the input and hidden layer, aim at re-estimating statistics on word occurrences so as to fit them to the target domain. In LINs, only the weights between the adaptation and hidden layers are updated and the others are fixed during training. In a similar manner, linear hidden networks (LHNs) [7], which insert an adaptation layer between the hidden and output layers, aim to adjust contextual phenomena to the target domain. Note that the parameters of NNLMs are sequentially estimated in the aforementioned methods. This is the reason why NNLMs tend to overfit to a small amount of data on the target domain. In contrast, the proposed method is able to avoid overfitting, because the weights for the domain-specific elements and those for the domain-shared elements are jointly estimated.

Feature augmentation for domain adaptation attempt to utilize external auxiliary features into RNNLMs. The one-hot vector representing the domains and the posterior probability of topic models are commonly used as external features [6, 7]. Frustratingly easy domain adaptation (FEDA) [8], which is one of the feature-augmentation approaches of domain adaptation, aims at augmenting features to two parts; the domain-shared part and the domain-specific part, and obtaining proper weights for each part by applying any standard training algorithms. The proposed method is inspired by the technique of FEDA. While FEDA augments the input features, the proposed method attempts to augment the vector representations of words and contexts, which enables the models capturing the dynamic alternation of the domain-shared and domain-specific phenomena.

3. RNNLMS

RNNLMs [1] have recurrent connections in the hidden layers that enable propagation of contextual information. Assume that the vocabulary size is V and the word at the time t denoted by $\mathbf{w}(t) \in \mathbb{R}^V$, which is represented by 1-of-K encoding. The matrix $\mathbf{L} \in \mathbb{R}^{E \times V}$ maps the word $\mathbf{w}(t)$ into the low dimensional continuous representation $\mathbf{p}(t) \in \mathbb{R}^E$

$$\mathbf{p}(t) = \mathbf{L}\mathbf{w}(t). \tag{1}$$

The hidden layer output $\mathbf{h}(t) \in \mathbb{R}^H$ is computed

$$\mathbf{h}(t) = f\left(\mathbf{p}(t), \mathbf{h}(t-1); \mathbf{U}, \mathbf{R}\right), \tag{2}$$

where H is the unit size of a hidden layer. $\mathbf{U} \in \mathbb{R}^{H \times E}$ denotes the weight parameters of the input connections, $\mathbf{R} \in \mathbb{R}^{H \times H}$ denotes the weight parameters of the recurrent connections, and $f(\cdot)$ represents an element-wise nonlinear function, such as the sigmoid or hyperbolic tangent functions. To capture a longer context than

RNNs, long short term memory (LSTM) cells [2], which have a constant error carousel $\mathbf{c}$ and three gates, namely, the input gate $\mathbf{i}$, forget gate $\mathbf{f}$, and output gate $\mathbf{o}$, are introduced into a hidden layer $\mathbf{h}(t)$. Given the context $\mathbf{h}(t)$, the RNNLMs yield the probability distribution $\mathbf{y}(t) \in \mathbb{R}^V$, which represents the probability of occurrence of the subsequent word as

$$\mathbf{y}(t) = \mathrm{softmax}\left(\mathbf{V}\mathbf{h}(t)\right), \tag{3}$$

where $\mathbf{V} \in \mathbb{R}^{V \times H}$ denotes the weight parameters for regression, and $\mathrm{softmax}\left(\mathbf{z}\right)_i = e^{z_i} / \sum_j e^{z_j}$. The RNNLMs are usually trained using truncated back propagation through time (BPTT).

4. RNNLMS WITH DOMAIN-SHARED AND DOMAIN-SPECIFIC REPRESENTATIONS

To capture the dynamic alternation of the domain-shared and domain-specific phenomena inside an RNNLM, the vector representations of words and contexts are simply extended into vectors that comprise domain-shared and domain-specific elements. The domain-specific elements consist of those of the source domain and the target domain separately and the elements are activated according to the domains, whereas the domain-shared elements are always activated. The parameters of the extended RNNLMs can be estimated without obscuring the domain-shared and domain-specific statistics using the standard algorithm for training RNNLMs.

With the proposed method, domain-shared and domain-specific representations are incorporated into the projection layer $\mathbf{p}(t)$ (Eq. (1)) and hidden layer $\mathbf{h}(t)$ (Eq. (2)). Figure 1 illustrates (a) the conventional RNNLMs and (b) the RNNLMs with domain-shared and domain-specific representations. With respect to the projection layer outputs, the continuous vector representation of a word $\mathbf{p}(t)$ is expanded into the three times larger vector representation $\mathbf{p}'(t)$ as

$$\mathbf{p}'(t) = \left[\mathbf{p}(t)^{\mathrm{T}} \;\; \mathbf{p}_S(t)^{\mathrm{T}} \;\; \mathbf{p}_T(t)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{4}$$

where $\mathbf{p}_d(t)$ $(d \in \{S, T\})$ equals $\mathbf{p}(t)$ if the domain of the current session is d and $\mathbf{0}_E$ otherwise. The parameters $\mathbf{U}$ for RNNLMs are also expanded to $\mathbf{U}' \in \mathbb{R}^{H \times 3E}$. The context vector $\mathbf{h}(t)$ is expanded into $\mathbf{h}'(t)$ in the same manner. In this case, $\mathbf{h}(t)$ is expanded in computing Eq. (3) as

$$\mathbf{h}'(t) = \left[\mathbf{h}(t)^{\mathrm{T}} \;\; \mathbf{h}_S(t)^{\mathrm{T}} \;\; \mathbf{h}_T(t)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{5}$$

where $\mathbf{h}_d(t)$ $(d \in \{S, T\})$ equals $\mathbf{h}(t)$ if the domain of the current session is d and $\mathbf{0}_H$ otherwise. The parameter $\mathbf{V}$ is expanded to $\mathbf{V}' \in \mathbb{R}^{V \times 3H}$. The network given source domain data and that given target domain data are shown in Figs. 1(c) and (d), respectively.

RNNLMs with domain-shared and domain-specific representations are trained using the training data including the source domain dataset $\mathcal{D}_S = \{s_S^{(1)}, s_S^{(2)}, \cdots s_S^{(N_S)}\}$ and target domain dataset $\mathcal{D}_T = \{s_T^{(1)}, s_T^{(2)}, \cdots s_T^{(N_T)}\}$, where $s_d^{(n)}$ denotes the n-th session (i.e., word sequence) in the domain $d \in \{S, T\}$. The models minimize a cross-entropy loss function with ℓ^2-regularization term given by

$$J(\Theta) = \sum_{d \in \{S,T\}} \sum_{n=1}^{N_d} \frac{1}{T_n} \sum_{t=1}^{T_n} \mathbf{d}_n(t) \log \mathbf{y}_n(t) + \frac{\beta}{2} \left|\left|\Theta\right|\right|_2^2, \tag{6}$$

where $\mathbf{d}_n(t)$ denotes a 1-of-K representation of the subsequent $(t+1)$ word in the n-th session, Θ denotes the parameters in RNNLMs, and β denotes the coefficient of ℓ^2-regularization term.

Table 1. *Setups for dataset*

	training		dev.	test
Dataset (Domain)	CSJ (source)	NTT (target)	NTT (target)	NTT (target)
# of sessions	967	40	8	8
vocab size	52 562	5 971	2 383	2 326
# of words	3.47M	0.15M	0.03M	0.03M
# of utterance	353 919	20 176	4 748	4 646

Table 2. *Hyperparameters for RNNLMs*

# of projection layer units E	100, 200, 300
# of hidden layer units H	50, 100, 150
initial learning rate α	0.1
the coefficient of ℓ^2-regularization term β	1.0×10^{-6}

5. SPEECH RECOGNITION EXPERIMENT

Experimental comparisons were conducted using lecture talks from the corpus of spontaneous Japanese (CSJ) [12] as the source domain data and multiparty conversations on specific topics as the target domain data. The multiparty conversations were recorded and transcribed into texts at the NTT Communication Science Laboratory. In the NTT dataset, four to six participants held a discussion on a specific topic for approximately 16 minutes. The mean of the utterance lengths in the NTT corpus was 7.6 words and the median of the utterance lengths was 3.0 words, while the mean and median in the CSJ corpus were 20.8 words and 13.0 words, respectively. Table 1 lists the details of the dataset.

The following language models were evaluated:

- 3-gram language models with Kneser-Ney smoothing [13],
- conventional LSTMLMs,
- LSTMLMs with adaptation layers, and
- **LSTMLMs with domain-shared and domain-specific representations.**

It should be noted that the *LSTMLMs* are trained only on the target domain and is regarded as the baseline model. The LSTMLMs trained using the data in both the source and target domain are referred to as "*data augmented LSTMLMs*." After training the *data augmented LSTMLMs*, an adaptation layer is inserted between the hidden and output layer. This model is referred to as "*data augmented LSTMLMs + LHNs*." Three ways of LSTMLMs with domain-shared and domain-specific representations are evaluated:

1. *LSTMLMs w/ shared and specific repr. (in)* denotes the LSTMLMs with domain-shared and domain-specific representations of words;
2. *LSTMLM w/ shared and specific repr. (out)* denotes the LSTMLMs with domain-shared and domain-specific representations of contexts; and
3. *LSTMLM w/ shared and specific repr. (in, out)* denotes the LSTMLMs with domain-shared and domain-specific representations of words and contexts.

The hyperparameters are listed in Table 2. All LSTMLMs have 5971 units in the output layer, irrespective of whether training data are augmented. The models are trained by the stochastic gradient descent and the learning rate is decayed by half when the ratio of the entropy on the validation data at the epoch τ to that at $\tau - 1$ is less

Table 3. *Validation set and test set perplexities (PPLs) and word error rates (WERs) for LSTMLMs trained under several settings.*

Models	Training data CSJ	Training data NTT	E	H	Valid PPLs	Test PPLs	WERs
w/o rescoring		✓			–	–	20.7
3-gram		✓			74.09	73.33	–
LSTMLMs		✓	300	150	56.60	56.33	20.4
data augmented LSTMLMs	✓	✓	300	150	45.43	45.82	19.7
data augmented LSTMLMs + LHNs	✓	✓	300	150	46.51	46.75	19.5
LSTMLMs w/ shared and specific repr. (in)	✓	✓	300	100	48.37	48.73	19.7
LSTMLMs w/ shared and specific repr. (out)	✓	✓	100	100	**34.75**	**35.01**	**19.3**
LSTMLMs w/ shared and specific repr. (in, out)	✓	✓	100	100	42.15	42.38	19.5

than a certain threshold. When the model training and evaluation of each session terminate, the history of the recurrent layer is flushed. We used a variant of LSTM referred to as "coupled input and forget gate" described in [14] and originally proposed in the context of gated recurrent units (GRU) [15], instead of the vanilla LSTM. The forget gate $\mathbf{f}$ in this variant of LSTM is computed as $\mathbf{1} - \mathbf{i}$, which reduces the complexity of the networks and might make the model robust despite resource limitations.

5.1. Perplexity evaluation

Table 3 lists the validation and test set perplexities for the LSTMLMs evaluated. The *data augmented LSTMLMs* reduced the perplexity averaged over sessions by approximately 18% compared with the *LSTMLM*. In this case, *data augmented LSTMLM* reduced the perplexity for all sessions in the test set. Since the LSTMs make it possible to deal with complicated phenomena, just augmenting the training data is dominant and LHNs did not help to improve the performance. The *LSTMLMs w/ shared and specific repr. (out)* and the *LSTMLMs w/ shared and specific repr. (in, out)* yielded an improvement in perplexity reduction over the *data augmented LSTMLM*. In this case, the number of parameters in *data augmented LSTMLM*, *LSTMLMs w/ shared and specific repr. (out)*, and *LSTMLMs w/ shared and specific repr. (in, out)* were about 2.96M, 2.47M, and 3.74M, respectively. *LSTMLM w/ shared and specific repr. (out)* could yield a significant reduction in perplexity over the *data augmented LSTMLM* without increasing model complexity. It indicated that regularization based on frustratingly easy domain adaptation in RNNLMs contributed more to the improvement than just increasing the number of parameters. Comparison of the three variations of the LSTMLMs with domain-shared and domain-specific representations indicated that augmenting the statistics on the contexts $\mathbf{h}$ as opposed to word embedding $\mathbf{p}$ contributed to the improvement of LSTMLMs.

To investigate the effect of utilizing the domain-shared and domain-specific representations jointly on prediction, the "*LSTMLMs w/ specific repr. (out)*", which only augmented the contexts $\mathbf{h}(t)$ as $\mathbf{h}'(t) = \left[\mathbf{h}_S(t)^{\mathrm{T}} \ \mathbf{h}_T(t)^{\mathrm{T}}\right]^{\mathrm{T}}$, were compared with *LSTMLMs w/ shared and specific repr. (out)*. Table 4 lists the test set perplexities. While *LSTMLMs w/ specific repr. (out)* reduced the perplexity compared with *data augmented LSTMLMs*, *LSTMLMs w/ shared and specific repr. (out)* outperformed *LSTMLMs w/ specific repr. (out)*. The linear interpolation between *data augmented LSTMLMs* and *LSTMLMs w/ specific repr. (out)*, which was regarded as the integration of domain-shared and domain-specific model over the probability of next word, still yielded higher perplexity than *LSTMLMs w/ shared and specific repr. (out)*. These results indicated that considering the both domain-shared and domain-specific representations jointly on prediction helped to improve the reduction in perplexity.

Table 4. *Perplexities (PPLs) for data augmented LSTMLMs, LSTMLMs w/ specific repr. (out) and LSTMLMs w/ shared and specific repr. (out).*

Models	Test PPLs
(1) data augmented LSTMLMs	45.82
(2) LSTMLMs w/ specific repr. (out)	43.45
linear interpolation between (1) and (2)	41.65
LSTMLMs w/ shared and specific repr. (out)	**35.01**

5.2. 500-best rescoring evaluation

A WFST-based speech recognizer [16] yielded a 500-best list for each utterance to evaluate ASR accuracy using 500-best rescoring. In this case, the acoustic model considered is a 6-layer fully-connected neural network comprising of an input layer with 418 units, hidden layers with 2048 units, and an output layer with 3874 units, and the language model is 3-gram with Kneser-Ney smoothing. Each hypothesis in the 500-best list was rescored using the LSTMLMs including the *LSTMLMs w/ shared and specific repr. (in, out)*. The best-rescored hypothesis of an utterance was used as the context for the subsequent utterance to reduce the computational complexity. Table 3 lists the word error rates obtained from the LSTMLMs evaluated. Augmentation of training data and domain adaptation yielded improvements in word error rates over the *LSTMLMs* by 0.9 points when using 500-best lists. The *LSTMLMs w/ shared and specific repr. (out)* steadily yielded improvements in word error rates over *data augmented LSTMLMs + LHNs* by 0.2 points, as well as the LSTMLMs with domain-shared and domain-specific representations yielded lower perplexity than the LSTMLMs with data augmentation and domain adaptation.

6. CONCLUSION

We proposed a method of effectively utilizing external resources based on LSTMLMs with domain-shared and domain-specific representations into a single model. The proposed language model domain adaptation is capable of capturing the commonalities and discrepancies between the source and target domains without obscuring them. Experimental comparison using multiparty conversation data demonstrated that the proposed method yielded improvements in the perplexity and word error rate over the simple augmentation of training data and model adaptation based on LHNs. Consequently, the proposed method reduced the perplexity by 38% and word error rates of 1.1 points from the LSTMLM without domain adaptation.

7. REFERENCES

[1] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH 2010*, Sept., pp. 1045–1048.

[2] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney, "LSTM neural networks for language modeling," in *Proc. INTERSPEECH 2012*, Sept., pp. 194–197.

[3] Robert C. Moore and William D. Lewis, "Intelligent selection of language model training data," in *Proc. ACL 2010*, July, pp. 220–224.

[4] Ryo Masumura, Seongjun Hahm, and Akinori Ito, "Training a language model using webdata for large vocabulary japanese spontaneous speech recognition," in *Proc. INTERSPEECH 2011*, Aug., pp. 1465–1468.

[5] Junho Park, Xunying Liu, Mark J. F. Gales, and Philip C. Woodland, "Improved neural network based language modelling and adaptation," in *Proc. INTERSPEECH 2010*, Sept., pp. 1041–1044.

[6] X. Chen, T. Tan, Xunying Liu, Pierre Lanchantin, M. Wan, Mark J. F. Gales, and Philip C. Woodland, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *Proc. INTERSPEECH 2015*, Sept., pp. 3511–3515.

[7] S. Deena, M. Hasan, M. Doulaty, O. Saz, and T. Hain, "Combining feature and model-based adaptation of rnnlms for multi-genre broadcast speech recognition," in *Proc. INTERSPEECH 2016*, Sept., pp. 2343–2347.

[8] Hal Daumé III, "Frustratingly easy domain adaptation," in *Proc. ACL 2007*, June.

[9] Ryo Masumura, Taichi Asami, Takanobu Oba, Hirokazu Masataki, Sumitaka Sakauchi, and Akinori Ito, "Combinations of various language model technologies including data expansion and adaptation in spontaneous speech recognition," in *Proc. INTERSPEECH 2015*, Sept., pp. 463–467.

[10] Anoop Deoras, Tomas Mikolov, Stefan Kombrink, Martin Karafiát, and Sanjeev Khudanpur, "Variational approximation of long-span language models for lvcsr," in *Proc. ICASSP 2011*, May, pp. 5532–5535.

[11] Stefan Kombrink, Tomas Mikolov, Martin Karafiát, and Lukás Burget, "Recurrent neural network based language modeling in meeting recognition," in *Proc. INTERSPEECH 2011*, Aug., pp. 2877–2880.

[12] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara, "Spontaneous speech corpus of japanese," in *Proc. LREC 2000*, May.

[13] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP 1995*, May, pp. 181–184.

[14] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Networks & Learning Systems*, vol. PP, no. 99, pp. 1–11, 2016.

[15] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP 2014*, Oct., pp. 1724–1734.

[16] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient wfst-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech & Lang. Process.*, vol. 15, no. 4, pp. 1352–1365, 2007.