INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Combining Feature and Model-Based Adaptation of RNNLMs for Multi-Genre Broadcast Speech Recognition

Salil Deena, Madina Hasan, Mortaza Doulaty, Oscar Saz and Thomas Hain

Speech and Hearing Research Group, The University of Sheffield, UK
{s.deena, m.hasan, m.doulaty, o.saz, t.hain}@sheffield.ac.uk

Abstract

Recurrent neural network language models (RNNLMs) have consistently outperformed n-gram language models when used in automatic speech recognition (ASR). This is because RNNLMs provide robust parameter estimation through the use of a continuous-space representation of words, and can generally model longer context dependencies than n-grams. The adaptation of RNNLMs to new domains remains an active research area and the two main approaches are: feature-based adaptation, where the input to the RNNLM is augmented with auxiliary features; and model-based adaptation, which includes model fine-tuning and introduction of adaptation layer(s) in the network. This paper explores the properties of both types of adaptation on multi-genre broadcast speech recognition. Two hybrid adaptation techniques are proposed, namely the fine-tuning of feature-based RNNLMs and the use of a feature-based adaptation layer. A method for the semi-supervised adaptation of RNNLMs, using topic model-based genre classification, is also presented and investigated. The gains obtained with RNNLM adaptation on a system trained on 700h. of speech are consistent using both RNNLMs trained on a small (10M words) and large set (660M words), with 10% perplexity and 2% word error rate improvements on a 28.3h. test set.

Index Terms: RNNLM, LM adaptation, multi-domain ASR

1. Introduction

Language models (LMs) play a key role in modern ASR and machine translation systems as they ensure that the output respects the pattern of the language in question. n-gram LMs dominated ASR for decades until RNNLMs [1] were introduced and found to give significant gains in performance. It is found that n-gram LM and RNNLM contributions are complementary and state-of-the-art ASR systems involve interpolation between the two types of models [1, 2, 3, 4, 5, 6, 7, 8, 9].

In automatic speech recognition, language context is generally heavily influenced by the domain, which can include topic, genre and speaking style. RNNLMs trained on a text corpus provide an implicit modelling of such contextual factors. However, it has been found that domain adaptation of RNNLMs to small amounts of matched in-domain text data provide significant improvements in both perplexity (PPL) and word error rate (WER) [10, 2, 3, 11, 5]. RNNLM adaptation can be categorised as either feature-based [2, 5] or model-based [10, 3, 11]. The former involves augmenting the input to the RNNLM with auxiliary features that encode domain information whilst the latter involves adapting the network to the new domain. Model-based RNNLM adaptation can either involve fine-tuning, which involves further training the RNNLM with matched in-domain data or the introduction of adaptation layer(s) to adapt the network to new domains.

Whilst feature-based RNNLM adaptation was shown to outperform domain fine tuning [5], it is required that the auxiliary features be known at the time of model training and thus can be inflexible, as it requires for the whole model to be re-trained should altered features become available. That in turn can be inconveninent as training an RNNLM on large amounts of data can take several days or even weeks to complete. Domain fine-tuning somehow addresses such limitation as the RNNLM can be fine-tuned using newly available domain-specific data and do not require retraining of the whole RNNLM. However, the shared information between domains is not properly modelled. A combination of feature and model adaptation can thus provide the best solution in many instances. This paper provides a detailed comparison of both types of adaptation on RNNLMs trained on both small and large text corpora and proposes novel techniques for RNNLM adaptation, including the linear hidden network (LHN) [22] adaptation layer as well as hybrid adaptation methods, and are evaluated on a broadcast media transcription task [12].

2. Recurrent Neural Network LMs

Recurrent Neural Network Language Models were introduced in [13] and include a recurrent layer which can represent the full history $h_i =< w_{i-1}, \ldots, w_1 >$ for word w_i using a concatenation of word w_{i-1} and the remaining context vector v_{i-2} from the previous time step. Each word w_i is represented using a 1-of-K encoding. The main advantages of RNNLMs over n-gram language models are: 1) it can represent the full, non-truncated history of words in an utterance and 2) it provides a continuous representation of the history and thus does not suffer from sparsity issues of n-gram LMs as some contexts might not occur in the data, and approximation techniques such as back-off are required [14, 15]. An out-of-vocabulary (OOV) node [10, 3, 5] can be included at the input to represent any input word that is not in the chosen vocabulary, and an out-of-shortlist (OOS) node [10, 3, 5] can be included at the output to represent any word not in a shortlist vocabulary. The main purpose of the latter is to reduce the computational cost at the output layer by limiting the vocabulary to the most frequent words. A further auxiliary feature vector f can be provided as input to the network, in order to allow for feature-based adaptation [2, 5].

The LM probability for the next word $P(w_{i+1}|w_i, v_{i-1})$ is computed as follows. A full history vector is obtained by concatenating w_i and the hidden (recurrent) layer activation from the previous time step, v_{i-1}. The hidden layer takes the two inputs and produces a new representation of the history, v_i using a non-linear sigmoid activation. This activation is then input to the softmax activation function at the output layer to produce normalised RNNLM probabilities. Moreover, the activa-

http://dx.doi.org/10.21437/Interspeech.2016-480

tion from the hidden layer is also returned to the input layer, as it encodes the word history, and is used to compute the probability for the following word. This is illustrated in Figure 1.

RNNLM training is performed using the back propagation through time (BPTT) algorithm [16], where the error is back-propagated through the recurrent connection for a specific number of time steps. The most expensive computation in RNNLM is the output softmax layer, which involves normalising the probabilities over the whole output vocabulary. This can be very costly when using cross entropy (CE) training with typical ASR tasks, which typically involve several thousand words in the output layer and several million words in the training corpus. In order to limit this computational effort, various approximation strategies have been developed. These include the hierarchical softmax (HS) [17], noise contrastive estimation (NCE) [18] and class-based approximations [19]. In this paper, the approach proposed by Chen *et al.*[4] is used, with GPU-based mini-batch training using spliced sentence bunch, allowing full softmax computation of the output using CE training.

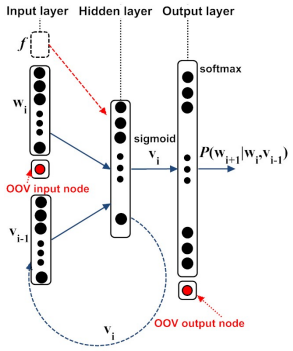

Figure 1: *Feature-based RNNLM with OOS and OOV nodes.*

3. Multi-Genre Broadcast Challenge Data

The experiments in this paper make use of the data provided by the British Broadcasting Corporation (BBC) for the Multi-Genre Broadcast (MGB) challenge 2015 [12]. Task 1 of the challenge involved participants having to perform the automatic transcription of a set of BBC shows. These shows were chosen to cover the multiple genres in broadcast TV, categorised in terms of 8 genres: advice, children's, comedy, competition, documentary, drama, events and news. Acoustic Model (AM) training data was fixed and limited to more than 2,000 shows, broadcast by the BBC during 6 weeks in April and May of 2008. The development data for the task consisted of 47 shows that were broadcast by the BBC during a week in mid-May 2008. The numbers of shows and the associated broadcast time for training and development data are shown in Table 1.

	Train		Development	
Genre	Shows	Time	Shows	Time
Advice	264	193.1h.	4	3.0h.
Children's	415	168.6h.	8	3.0h.
Comedy	148	74.0h.	6	3.2h.
Competition	270	186.3h.	6	3.3h.
Documentary	285	214.2h.	9	6.8h.
Drama	145	107.9h.	4	2.7h.
Events	179	282.0h.	5	4.3h.
News	487	354.4h.	5	2.0h.
Total	2,193	1580.5h.	47	28.3h.

Table 1: *Amount of training and development data.*

Additional data was available for Language Model (LM) training in the form of subtitles from shows broadcast from 1979 to March 2008, with a total of 650 million words, and referred to as $LM1$. The subtitles from the 2,000+ shows for acoustic modelling could also be used for LM training, referred to as $LM2$. Statistics for these two sets can be seen in Table 2.

The development data was used as the evaluation set in order to provide fair comparison with previous work [5, 20]. For language model experiments, the $LM2$ data was partitioned into a training and development set by selecting 90% of text for each programme for training and the remaining 10% for development, after shuffling the lines for each programme.

Subtitles	#sentences	#words	#unique words
$LM1$ (1979-2008)	72.9M	648.0M	752,875
$LM2$ (Apr/May '08)	633,634	10.6M	32,304

Table 2: *Language model data.*

The rest of the paper will deal with RNNLM adaptation to multiple domains in the context of the MGB challenge.

4. Feature-Based RNNLM Adaptation

In feature-based RNNLM adaptation, a feature vector f is appended to the input of the RNNLM as shown in Figure 1. Two features are used in this work, which are as follows.

4.1. Genre 1-hot Auxiliary Codes

Genre information can be represented using a 1-of-K encoding, with K being 8. Given that ground truth genre information is available for each show, the latter can be input to the RNNLM as a feature vector, at both training and test times.

4.2. LDA Auxiliary Features

Latent Dirichlet Allocation (LDA) [21] is a generative model that allows text data to be represented by a set of unobserved topics. Term frequency-inverse document frequency (TF-IDF) vectors are computed on $LM2$ training text data, which are used to train LDA models. LDA features are then obtained by computing Dirichlet posteriors over the topics for each show. LDA features were found to giver better performance than genre features when used for RNNLM adaptation on the MGB data [5]. This is due to LDA features providing a finer representation of domain than genre auxiliary codes, through the use of a continuous feature space and over a larger number of latent topics.

5. Model-Based RNNLM Adaptation

5.1. Model Fine-tuning

Model fine-tuning is one way of adapting a RNNLM to a specific genre, and it involves further training the RNNLM on genre-specific data, resulting in a separate model per genre.

5.2. LHN Adaptation Layer

An adaptation layer can be cascaded in the network and then fine-tuned by only updating the weights connecting the adaptation layer and the next layer. This was done for feed-forward neural network LMs (NNLM), where the adaptation layer was cascaded between the projection layer and the hidden layer [10]. A projection layer is necessary for NNLMs because $n-1$ inputs, corresponding to n-gram contexts, need to be projected to a lower dimension, before being fed to the hidden layer. In RNNLMs, the projection layer is not necessary as only the recurrent vector is fed as input to the hidden layer [13]. In this work, the adaptation layer is cascaded between the hidden layer and the output layer as shown in Figure 2. The adaptation layer has a linear activation and thus provides a linear transform to the hidden layer. This is equivalent to the linear hidden network (LHN) [22] transform that has been applied to DNN acoustic

models. The weights connecting the hidden and adaptation layers are initialised as the identity matrix, thus providing an equivalent network to the unadapted RNNLM. At the time of fine-tuning, only the weights connecting the adaptation to the output layer are updated. To the best of our knowledge, we are the first to apply LHN adaptation to RNNLMs.

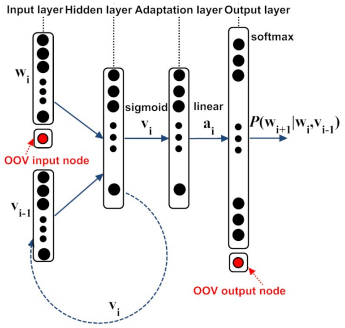

Figure 2: *RNNLM with LHN adaptation layer.*

6. Hybrid RNNLM Adaptation

6.1. Fine-tuning Feature-Based RNNLM

Feature-based RNNLMs with LDA features can be further fine-tuned to each genre by further training the models on genre-specific text. This is one way in which topic and genre information can be leveraged effectively for RNNLM adaptation.

6.2. Feature-Based RNNLM with Adaptation Layer

One disadvantage of the LHN adaptation layer fine-tuning of RNNLMs is that overfitting can happen if the amount of genre-specific data is small. It was shown in [11] that the adaptation layer can be recast from providing a multiplicative transform (as in the case of the LHN transform) to an additive transform, by using a domain vector d in the form of a 1-of-K encoding, as input to the adaptation layer. In our case, this is equivalent to using genre 1-hot vectors as input to the adaptation layer. Such a model can also include auxiliary features such as LDA, as input to the hidden layer, as shown in Figure 3.

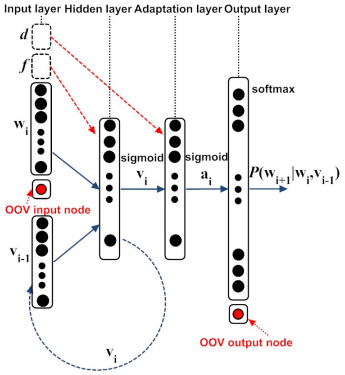

Figure 3: *RNNLM with feature-based adaptation layer.*

7. Semi-supervised RNNLM Adaptation

In the MGB challenge, genre information is only available for the transcripts of the acoustic data ($LM2$) and not for the separate language model ($LM1$) subtitle text. $LM1$ text is much richer with 650M words. In order to apply genre adaptation when using $LM1$ text, genre labels need to be automatically derived. LDA features allow for a good classification of genre when using support vector machine (SVM) classification, as was verified on the development set. An experiment was performed where a SVM was trained to predict genre labels from LDA features extracted on the acoustic text training set and tested on the official development set. It was found that 1024 topics gave the best result with a classification accuracy of 94.79%. The same model is used further to predict the genre labels for the subtitle ($LM1$) text in order to provide the genre feature input for $LM1\&LM2$ RNNLMs.

8. Experiments and Results

A vocabulary of 200k words was chosen from all the words in the $LM2$ text (87k) and augmented with the most frequently occuring words in $LM1$ for training the baseline 4-gram LM. For acoustic modelling, 700h. of speech was selected from the training set based on word matching error rate (WMER) and confidence scores [23]. The acoustic models consisted of *Bottleneck* DNN-GMM-HMM trained using TNet [24] and HTK [25] toolkits. The *Bottleneck* system used a DNN for extracting 26 features. The DNN took as input 15 contiguous log-filterbank frames and consisted of 4 hidden layers of $1,745$ neurons plus the 26-neuron *Bottleneck* layer, and an output layer of $8,000$ triphone state targets. State-level Minimum Bayes Risk (sMBR) [26, 27] was used as the target function for training the DNN. Feature vectors for training the GMM-HMM systems were 65-dimensional, including the 26 dimensional *Bottleneck* features, as well as 13 dimensional PLP features together with their first and second derivatives. GMM-HMM models were trained using 16 Gaussian components per state, and around 8k distinct triphone states. More details on our system for the MGB challenge can be found in [23].

Our baseline 4-gram language model was trained using $LM1\&LM2$ text with the SRILM toolkit [28] using the 200k vocabulary. Our baseline RNNLM was trained using $LM1\&LM2$ text with a 60k vocabulary for the input word list and a 50k vocabulary for the output word list. Both the 60k and 50k wordlists were obtained by shortlisting the 200k vocabulary based on most frequent words.

Decoding with *Bottleneck* systems was performed in three stages; in a first stage, lattices were generated using a 2-gram LMs, followed by lattice rescoring with a 4-gram LM to generate new lattices. The 4-gram lattices were further rescored using the RNNLM using the n-gram approximation lattice rescoring method described in [29], with n set to 6 as this was found to give optimal results. In addition, n-best list rescoring was performed by first converting the lattices to n-best lists, with n being 100, followed by 1-best computation.

The baseline results, as well as results obtained using adaptation of RNNLMs trained on $LM2$ and $LM1\&LM2$ text and scored using the official MGB scoring package [12], are shown in Table 3. All RNNLMs in our experiments have 512 nodes for the hidden layer and where applicable, 512 nodes for the adaptation layer. It was found that n-best list rescoring gives an improvement of 0.2% in the WER over lattice rescoring when using $LM1\&LM2$ RNNLMs. As a result, n-best list rescoring was used for all RNNLM adaptation experiments. The interpolation weight of $LM1\&LM2$ RNNLMs with the 4-gram baseline LM was set to 0.5 as this was found to give the lowest PPL on our development set. Similarly, an interpolation weight of 0.3 gives the lowest PPL for $LM2$ RNNLMs.

LDA auxiliary features are found to be more effective than genre 1-hot features for the adaptation of acoustic text ($LM2$) RNNLMs, similar to what has been reported in the literature [5, 20]. The number of LDA topics was varied from 10 to 150

	Genre →	Adv.	Child.	Comed.	Compet.	Docum.	Dram.	Even.	News	Global	
System	Adaptation	WER								PPL	WER
$LM1\&LM2$ 4-gram and RNNLM baselines											
4-gram	None	24.6	30.4	43.5	25.8	28.0	41.5	34.1	15.7	100.1	30.1
4-gram+RNNLM interp (lattice rescoring)	None	23.8	29.4	43.1	25.5	27.3	41.5	32.9	14.8	88.6	29.4
4-gram+RNNLM interp (n-best rescoring)	None	23.7	29.2	43.2	25.0	26.9	41.7	32.7	14.5	88.6	29.2
$LM1\&LM2$ 4-gram + $LM2$ RNNLM (0.3 interp) with RNNLM adaptation											
RNNLM Baseline	None	24.2	29.8	43.6	25.5	27.7	42.2	33.3	14.9	93.7	29.8
Genre feat. at hidden layer	Feature	24.3	29.6	43.5	25.2	27.6	42.0	33.1	14.9	91.9	29.7
Genre fine-tuning	Model	24.3	29.6	43.4	25.3	27.5	41.6	33.2	14.8	90.6	29.6
Genre LHN adaptation layer fine-tuning	Model	24.1	29.5	43.3	25.2	27.6	41.7	33.1	15.0	90.4	29.6
Genre feat. at adaptation layer	Hybrid	23.9	29.6	43.5	25.3	27.4	42.0	33.2	14.9	90.7	29.6
LDA feat. at hidden layer	Feature	23.9	29.4	43.6	25.1	27.6	41.4	32.7	14.7	88.3	29.5
LDA feat. at hidden layer and genre fine-tuning	Hybrid	23.9	29.3	43.6	24.8	27.5	41.3	32.7	14.8	86.7	29.4
LDA feat. at hidden and genre feat. at adaptation layer	Hybrid	23.6	28.9	43.4	24.9	27.3	41.2	32.5	14.6	86.9	29.2
$LM1\&LM2$ 4-gram + $LM1\&LM2$ RNNLM (0.5 interp) with RNNLM adaptation											
RNNLM Baseline	None	23.7	29.2	43.2	25.0	26.9	41.7	32.7	14.5	88.6	29.2
Genre feat. at hidden layer	Feature	23.5	29.1	42.6	24.6	26.9	40.5	32.9	14.6	85.4	29.0
Genre fine-tuning	Model	23.6	28.9	42.7	24.5	26.9	41.2	32.5	14.3	82.2	29.0
Genre LHN adaptation layer fine-tuning	Model	23.4	28.8	42.6	24.6	26.9	41.2	32.4	14.2	81.9	28.9
Genre feat. at adaptation layer	Hybrid	23.1	28.6	42.4	24.2	26.5	40.4	32.6	14.3	83.4	28.7
LDA feat. at hidden layer	Feature	23.1	28.7	42.5	24.5	26.5	40.4	32.3	14.5	81.6	28.7
LDA feat. at hidden layer and genre fine-tuning	Hybrid	23.0	28.7	42.5	24.4	26.5	40.4	32.3	14.4	80.4	28.7
LDA feat. at hidden and genre feat. at adaptation layer	Hybrid	22.9	28.6	42.5	24.2	26.4	40.3	32.3	14.1	79.4	28.6

Table 3: *RNNLM baseline and adaptation results on MGB data.*

and it was found that 100 topics gives the best result, both in terms of test PPL and WER, when extracting LDA features from the reference text for each show. The number of topics was thus fixed to 100. It was shown in [5] that computing LDA features from the ASR output led to a degration of performance by about 0.1% when using 30 LDA topics. In contrast, we found that with 100 topics, the same overall WER result is obtained when using the reference and ASR output, with some minor variations within genres. The ASR output text is thus used to compute LDA features. It is interesting to note that the LDA-adapted $LM1\&LM2$ RNNLM with 100 topics also results in a substantial drop in WER of 0.5%. Moreover, small but significant gains are obtained with the genre fine-tuning of LDA-adapted RNNLMs. For the $LM2$ RNNLM, this hybrid adaptation leads to a global drop in WER from 29.5% to 29.4%.

The LHN adaptation layer fine-tuning is found to work well for RNNLMs, giving gains in terms of PPL over full model fine-tuning for both $LM2$ and $LM1\&LM2$ RNNLMs. With $LM1\&LM2$ RNNLM, there is also a drop in the global WER (28.9%) using the LHN adaptation layer compared to full model fine-tuning (29.0%). It is to be noted that such small gains are significant for language modelling.

Our experiments with the introduction of a domain-specific adaptation layer also corroborate with previous work on acoustic model DNN adaptation which showed that using an adaptation layer with additive bias adaptation (feature-based adaptation layer) works better than a multiplicative transform (LHN adaptation layer) [30]. Whilst the performance of both are comparable using $LM2$ RNNLMs with a WER of 29.6%, the results with $LM1\&LM2$ RNNLMs indicate that an additive transform performs better (WER 28.7%) compared to when using a multiplicative transform (WER 28.9%). This might be explained by the fact that a multiplicative transform is more prone to over-fitting, especially when the amount of in-domain data is limited (e.g. in genres such as comedy and drama).

The results show that LDA features can be used to classify genre quite accurately using SVMs and these genre labels are useful when training a $LM1\&LM2$ RNNLM in a semi-supervised fashion. Using those genre labels as input features to the adaptation layer leads to a drop in WER of 0.5% from 29.2% to 28.7%. RNNLMs trained with those LDA-derived genre labels at the adaptation layer give comparable results to using LDA features input to the hidden layer, for $LM1\&LM2$ RNNLMs with a WER of 28.7% for both, although combining the two inputs yields a further improvement to 28.6%.

Finally, the results show the complementarity between the two domain representations, namely topic representation derived from LDA features and genre representation provided for training, development and evaluation sets in the MGB challenge. Combining them, i.e. topic and genre at the hidden and adaptation layers respectively, gives the best results with a drop in WER of 0.6% from 29.8% to 29.2% using an RNNLM trained on $LM2$ text and from 29.2% to 28.6% using an RNNLM trained on $LM1\&LM2$ text.

9. Conclusions

In this work, various feature and model-based adaptation methods for RNNLMs have been compared and combined on multi-genre speech recognition. It was found that the two approaches can be complementary and combining them often improves performance. The use of a separate adaptation layer was investigated for genre adaptation, using either an additive or a multiplicative transform, with an additive transform giving better results. Finally, it was found that using topic and genre features together, lead to better results than when using either input on its own. In future work, the joint modelling of topic and language with RNNLMs will be investigated, in order to devise novel adaptation techniques for multi-domain ASR.

10. Acknowledgements and Data Access

The authors would like to thank Cambridge University for releasing the CUED RNNLM toolkit which this work builds on. The audio and subtitle data used for the experiments was distributed as part of the MGB Challenge (mgb-challenge.org) through a licence with the BBC. The CTM and scoring files can be accessed via DOI: 10.15131/shef.data.3141910. This work was supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

11. References

[1] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH'10: Proc. of the 11th Annual Conference of the International Speech Communication Association*, 2010, pp. 1045–1048.

[2] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model." in *SLT'12: Proc. of the IEEE workshop on Spoken Language Technologies*, 2012, pp. 234–239.

[3] T. Alumäe, "Multi-domain neural network language model," in *INTERSPEECH'13, 14th Annual Conference of the International Speech Communication Association*, 2013, pp. 2182–2186.

[4] X. Chen, Y. Wang, X. Liu, M. J. Gales, and P. C. Woodland, "Efficient GPU-based training of recurrent neural network language models using spliced sentence bunch." in *INTERSPEECH'14: Proc. of the 11th Annual Conference of the International Speech Communication Association*, 2014, pp. 641–645.

[5] X. Chen, T. Tan, X. Liu, P. Lanchantin, M. Wan, M. J. F. Gales, and P. C. Woodland, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *INTERSPEECH'15: Proc. of the 16th Annual Conference of the International Speech Communication Association*, 2015, pp. 3511–3515.

[6] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," in *ICASSP'15: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5391–5395.

[7] Y. Tachioka and S. Watanabe, "Discriminative method for recurrent neural network language models," in *ICASSP'15: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5386–5390.

[8] E. Arisoy, A. Sethy, B. Ramabhadran, and S. F. Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *ICASSP'15: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5421–5425.

[9] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[10] J. Park, X. Liu, M. J. F. Gales, and P. C. Woodland, "Improved neural network based language modelling and adaptation," in *INTERSPEECH'10: Proc. of the 11th Annual Conference of the International Speech Communication Association*, 2010, pp. 1041–1044.

[11] O. Tilk and T. Alumäe, "Multi-domain recurrent neural network language model for medical speech recognition," in *Human Language Technologies - The Baltic Perspective - Proceedings of the Sixth International Conference Baltic HLT 2014, Kaunas, Lithuania, September 26-27, 2014*, 2014, pp. 149–152.

[12] P. Bell, M. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Webster, and P. Woodland, "The MGB challenge: Evaluating multi–genre broadcast media transcription," in *ASRU'15: Proc. of IEEE workshop on Automatic Speech Recognition and Understanding*, Scottsdale, AZ, 2015.

[13] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." *INTERSPEECH'10: Proc. of the 11th Annual Conference of the International Speech Communication Association*, vol. 2, p. 3, 2010.

[14] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, Mar. 1987.

[15] R. Kneser and H. Ney, "Improved backing-off for M-gram language modeling," in *ICASSP'95: Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1995, pp. 181–184.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. Oct, pp. 533–536+, 1986.

[17] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *AISTATS'05: Proc. of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 246–252.

[18] X. Chen, X. Liu, M. J. F. Gales, and P. C. Woodland, "Recurrent neural network language model training with noise contrastive estimation for speech recognition." in *ICASSP'15: Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, 2015, pp. 5411–5415.

[19] T. Mikolov, S. Kombrink, L. Burget, J. Cernock, and S. Khudanpur, "Extensions of recurrent neural network language model." in *ICASSP'11: Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, 2011, pp. 5528–5531.

[20] P. C. Woodland, X. Liu, Y. Qian, C. Zhang, M. J. F. Gales, P. Karanasou, P. Lanchantin, and L. Wang, "Cambridge university transcription systems for the multi-genre broadcast challenge," in *ASRU'15: Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2015, pp. 639–646.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[22] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. de Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models." *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[23] O. Saz, M. Doulaty, S. Deena, R. Milner, R. W. M. Ng, M. Hasan, Y. Liu, and T. Hain, "The 2015 Sheffield system for transcription of multi–genre broadcast media," in *ASRU'15: Proc. of the IEEE Automatic Speech Recognition and Understanding workshop*, 2015.

[24] K. Vesely, L. Burget, and F. Grezl, "Parallel training of neural networks for speech recognition," in *INTERSPEECH'10: Proc. of the Annual Conference of the International Speech Communication Association*, 2010, pp. 2934–2937.

[25] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, L. Liu, G. Moore, J. J. Odell, D. G. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book version 3.4*. Cambridge University Engineering Department, 2006.

[26] M. Gibson and T. Hain, "Hypothesis Spaces For Minimum Bayes Risk Training In Large Vocabulary Speech Recognition," in *INTERSPEECH'06: Proc. of the Annual Conference of the International Speech Communication Association*, Pittsburgh, PA, 2006, pp. 2406–2409.

[27] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *INTERSPEECH'06: Proc. of the Annual Conference of the International Speech Communication Association*, 2012.

[28] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *ICSLP'02: Proc. of International Conference on Spoken Language Processing*, 2002.

[29] X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *ICASSP'14: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4908–4912.

[30] Y. Liu, P. Karanasou, and T. Hain, "An Investigation Into Speaker Informed DNN Front-end for LVCSR," in *ICASSP'15: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2015.