Recurrent Neural Network Language Model Adaptation for Multi-Genre Broadcast Speech Recognition and Alignment

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

***Abstract*—Recurrent neural network language models (RNNLMs) generally outperform n-gram language models when used in automatic speech recognition. Adapting RNNLMs to new domains is an open problem and current approaches can be categorised as either feature-based or model-based. In feature-based adaptation, the input to the RNNLM is augmented with auxiliary features whilst model-based adaptation includes model fine-tuning and the introduction of adaptation layer(s) in the network. In this paper, the properties of both types of adaptation are investigated on multi-genre broadcast speech recognition. Existing techniques for both types of adaptation are reviewed and the proposed techniques for model-based adaptation, namely the linear hidden network (LHN) adaptation layer and the K-component adaptive RNNLM, are investigated. Moreover, new features derived from the acoustic domain are investigated for RNNLM adaptation. The contributions of this paper include two hybrid adaptation techniques: the fine-tuning of feature-based RNNLMs and a feature-based adaptation layer. Moreover, the semi-supervised adaptation of RNNLMs using genre information is also proposed. The ASR systems were trained using 700h of multi-genre broadcast speech. The gains obtained when using the RNNLM adaptation techniques proposed in this work are consistent when using RNNLMs trained on an in-domain set of 10M words and on a combination of in-domain and out-of-domain sets of 660M words, with approx. 10% perplexity and 2% relative word error rate improvements on a 28.3h. test set. The best RNNLM adaptation techniques for ASR are also evaluated on a lightly supervised alignment of subtitles task for the same data, where the use of RNNLM adaptation leads to an absolute increase in the F–measure of 0.5%.**

***Index Terms*—speech recognition, RNNLM, language model adaptation, multi-domain ASR**

I. Introduction

LANGUAGE models (LMs) are a major component of automatic speech recognition (ASR) systems as they ensure that the output is consistent with the language in question. n-gram LMs were dominant until neural network language models (NNLMs) [1] and recurrent neural network language models (RNNLMs) [2] were introduced. Moreover, it has been found that n-gram LM and NNLM/RNNLM contributions are complementary and an interpolation between the two types of models [2], [3], [4], [5], [6], [7], [8], [9] usually lead to the best results. In this paper, we will be focussing on RNNLMs, which have the ability to model longer temporal dependencies than n-grams and NNLMs.

The authors are with the Speech and Hearing Research Group, Department of Computer Science, The University of Sheffield, Sheffield, UK. Email: {s.deena, m.hasan, m.doulaty, o.saz, t.hain}@sheffield.ac.uk

Manuscript received xx xx, 2017; revised xx xx, 2017.

In ASR, word context is generally heavily influenced by the domain, which is mostly characterised by the topic of the discourse and speaking style. Generally, RNNLMs provide an implicit modelling of domain. However, it has been found that the adaptation of NNLMs/RNNLMs to small amounts of matched in-domain text data can yield a decrease in both perplexity (PPL) and word error rate (WER) [10], [3], [4], [11], [6]. This work investigates adaptation strategies for RNNLMs in the context of both ASR and lightly supervised alignment (LSA).

LSA [12], [13] is a task which is related to speech transcription and involves aligning a given text to an audio recording by identifying words that match the audio and provide their timings. Typical alignment systems perform speech recognition with language models that are biased towards the target material using interpolation with a background language model, which is followed by matching the transcribed output to the input text. This paper explores the use of RNNLMs as biased language models, through the fine-tuning of RNNLMs to subtitle text, used in conjunction with a background n-gram language model biased towards the subtitles text. The use of domain adaptation for such RNNLMs allow for the ASR output to be better matched to the input text, which in turn leads to better alignment results.

The adaptation techniques proposed in this work are investigated on both the ASR and LSA tasks of the multi-genre broadcast challenge [14]. The structure of the paper is as follows. Section II reviews existing work, including RNNLMs and its variants as well as language model domain adaptation. Section III describes the multi-genre broadcast data used in this work. Section IV then explores RNNLM adaptation techniques for multi-genre broadcast data, where novel techniques are proposed. In section V, the RNNLM domain adaptation techniques are investigated for ASR whilst section VI investigates the RNNLM domain adaptation techniques for LSA. In section VII, the experimental setup is described and results are presented and discussed. Finally concluding remarks are given in section VIII.

II. Related Work

In this section, we first review RNNLMs, which is followed by a review of language model adaptation techniques, so as to better motivate the techniques proposed in this work.

A. Recurrent Neural Network LMs

RNNLMs 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}. A 1-of-K encoding is used to represent each word w_i. RNNLMs offer the following advantages over n-gram LMs: 1) the full, non-truncated history of words in an utterance can be modelled and 2) the continuous representation of the history means that they do not suffer from sparsity issues of n-gram LMs, which require the use of techniques such as back-off[15], [16], when some contexts do not occur in the data.

The structure of a RNNLM is shown in Figure 1. In order to represent any input word that is not in the chosen vocabulary, an out-of-vocabulary (OOV) node [10], [4], [6] can be included at the input. Similarly, an out-of-shortlist (OOS) node [10], [4], [6] can be included at the output. The aim is to reduce the computational cost at the output layer by limiting the vocabulary to the most frequent words. In order to allow for feature-based adaptation [3], [6], an auxiliary feature vector f can be provided as input to the network.

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 v_{i-1}, the hidden layer activation from the previous time step. The hidden layer takes this as input and produces a new representation of the history, v_i using a non-linear activation function (sigmoid in our case). This is then passed to the softmax activation function at the output layer to produce RNNLM probabilities. In order to compute the probability for the following word, the hidden layer activation is returned to the input layer, given that it encodes the word history and the process is repeated.

Back propagation through time (BPTT) [17] is used for training the RNNLM, which involves back-propagating the error through the recurrent connection for a specific number of time steps. The output softmax layer involves the most expensive computation when using cross entropy (CE) training, as it requires normalising the probabilities over the whole output vocabulary. Various approximation strategies have been proposed to address this. These include: noise contrastive estimation (NCE) [18], hierarchical softmax (HS) [19], and class-based structuring of the softmax layer [20]. We use the approach proposed by Chen *et al.*[5] with GPU-based mini-batch training using spliced sentence bunch, which allows full CE-training based softmax computation of the output, thus not relying on approximations to the softmax, which can compromise performance.

In this work, RNNLMs are interpolated with n-gram LMs, which have a larger vocabulary than the RNNLM. The words that are in the n-gram vocabulary and not in the RNNLM vocabulary are replaced by the `<UNK>` symbol and modelled using the OOV node. The OOV probabilities are then re-normalised according to the method described in [21]. Linear interpolation is used to interpolate the probabilities of the n-gram LM and RNNLM.

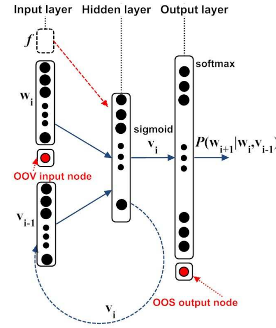


Fig. 1: *Feature-based RNNLM with OOV and OOS nodes.*

B. Review of Language Model Adaptation

Language model adaptation for specific domains is particularly important for ASR because the pattern of word sequences is heavily influenced by the subject matter. Natural language encodes both short and long term word sequence dependencies with often rapid changes in topic and theme. In addition, different domains involve relatively disjoint concepts with different word sequence statistics. In the context of multi-genre broadcast speech recognition, the domain is known beforehand and thus adapting a generic language model to a specific domain should help with the recognition rate.

1) n-gram Language Model Adaptation: n-gram language models are based on relative frequencies of n-gram events and according to [22] adaptation techniques can be broadly classified as:

- Model interpolation, where probabilities are combined either at the word or sentence level between a background language model and domain-specific language models [23].
- Constraint specification, which involve integrating multiple sources of information in the form of features using techniques such as exponential models using the maximum entropy criterion [24].
- Mixture language models, using topic information which are extracted from the underlying text data and used to determine the weights of each sub-model [25].

Neural network-based language models use a different architecture and as such, the adaptation strategies are different, as discussed next.

2) Neural Network Language Model Adaptation: NNLM/RNNLM adaptation can be broadly categorised as feature-based [3], [6] or model-based [10], [4], [11], [9]. Whilst the former augments the input with auxiliary features that encodes domain information, the latter adapts the network to the new domain. Model-based adaptation can be further categorised into: fine-tuning, which involves further training the language model on the in-domain data and; the introduction of adaptation layer(s) to the network.

In [10], [4], [26], [27], domain-specific adaptation layers were introduced, which took the form of either a multiplicative [10], [27] or an additive [4] transform of the neural network weights. In the case of a multiplicative transform, a new adaptation layer is cascaded between either the input and the hidden layer (Linear Input Network/LIN), between the hidden layer(s) and output layer (Linear Hidden Network/LHN) or as a linear transform to the output layer (Linear Output Network/LON). The layer is initialised to the identity matrix with a linear activation and at the time of adaptation, only the weights associated with the adaptation layer are updated, whilst the rest of the network is kept fixed. In addition, fine-tuning [28], [29] can also be used for adaptation, where the structure of the network is kept fixed, but the network is further trained on domain-specific text data.

3) Recurrent Neural Network Language Model Adaptation: Recurrent neural network language adaptation can also be model-based as in the case of NNLMs [11]. Feature-based RNNLM adaptation has also been proposed [3], [6], where an auxillary feature vector is appended at the input layer to encode the domain.

Feature-based RNNLM adaptation requires that the auxiliary features be known at the time of training and can be inflexible, requiring for the whole model to be re-trained should altered features become available. This is not practical as training an RNNLM on large amounts of data can take several days or weeks to complete, depending on the amount of data and computing infrastructure available. On the other hand, domain fine-tuning does not require retraining of the whole RNNLM as the RNNLM is only fine-tuned on newly available domain-specific data. In addition, it was shown by Chen *et al.*[6] that feature-based RNNLM adaptation can outperform domain fine-tuning in ASR. A combination of feature and model-based adaptation can provide a solution that is both flexible and effective at leveraging on shared information between domains.

In this paper, a detailed comparison of both types of adaptation is presented. This includes the linear hidden network (LHN) [30] adaptation layer, which was first proposed for RNNLM adaptation in [31]. Moreover, hybrid adaptation methods that combine the strengths of feature and model-based adaptation, are proposed and evaluated on a broadcast media transcription task [14].

III. Multi-Genre Broadcast Data

In this work, the data provided by the British Broadcasting Corporation (BBC) for the Multi-Genre Broadcast (MGB) challenge 2015 [14], is used in experiments to demonstrate the proposed methods. Task 1 of the challenge involved the automatic transcription of a set of BBC shows whilst Task 2 involved lightly supervised alignment. The BBC shows were chosen to cover the multiple genres in broadcast TV and were categorised as 8 genres, namely: advice, children's, comedy, competition, documentary, drama, events and news. For acoustic modelling (AM), the training data was fixed to more than 2,000 shows, broadcast by the BBC during 6 weeks, from April to May of 2008. The development data provided consisted of 47 shows that were broadcast by the BBC during a week in May 2008. Finally, evaluation data was released only at the time of the challenge. Table I shows the distribution of the data in terms of shows for the training, development and evaluation sets.

TABLE I: *MGB Data distribution.*

Dataset	Number of Shows	Broadcast Time
Training	2,193	1,580.4 h.
Development	47	28.4 h.
Evaluation	16	11.2 h.

Table II shows the numbers of shows and the associated broadcast time for the training and development sets across the 8 genres.

TABLE II: *Amount of training and development data for ASR Experiments.*

	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.

Language Model (LM) training was also released for the challenge. These were in the form of subtitles of BBC shows broadcast from 1979 to March 2008, with a total of 650 million words, and referred to as $LM1$ in this work. We also use the subtitles from the 2,000+ shows for acoustic modelling for LM training, and refer to it as $LM2$. Table III shows the statistics for these two sets.

TABLE III: *Language model data.*

Subtitles	#sentences	#words	#unique words
$LM1$ (1979-2008)	72.9M	648.0M	752.9K
$LM2$ (Apr/May '08)	633.6K	10.6M	32.3K

IV. RNNLM Adaptation for Multi-Genre Broadcast Media

A. 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. Feature-based RNNLM adaptation has proven to be effective at adapting RNNLMs for the MGB challenge task [6] as they augment the RNNLM with domain-specific information, thus providing an extra level of representation than text. Whilst previous work has investigated the use of genre and topic information derived from text [6], in this work the use of topic information derived from acoustic data is investigated. We also investigate whether text-based and acoustic-based topic information can be combined when used for RNNLM domain adaptation.

1) Genre Auxiliary Features: Genre information can be represented as a 1-hot vector, using a 1-of-K encoding. In MGB data, genre information is available for each show. Hence the genre 1-hot vectors can be input to the RNNLM as a feature vector both for training and test. The genre auxiliary codes can thus help to structure the diversity in the data by encoding explicit domain information in the model.

2) Text-based LDA Auxiliary Features: Latent Dirichlet Allocation (LDA) [32] allows text to be represented by a set of unobserved topics, through the use of generative probabilistic modelling. LDA models are first trained by extracting term frequency-inverse document frequency (TF-IDF) vectors on the text. LDA features are then derived by computing Dirichlet posteriors over the topics. In this work, the LDA features are extracted at show level. Chen *et al.*[6] showed that LDA features outperformed genre features when used for RNNLM adaptation on the MGB data. This could be because the LDA features provide a continuous feature space representation and over a larger number of latent topics, than can be achieved with genre 1-hot vectors.

One disadvantage when using text-based LDA features for domain adaptation is that it needs to be done in a two-pass decoding system. In the first pass decoding stage, a baseline ASR system needs to generate an ASR output, from which LDA features are extracted. The LDA features are then used in the second-pass decoding stage to give an improved transcription with a LDA-adapted RNNLM. Another disadvantage of this method is that the LDA features used for training the RNNLM are usually generated using ground truth text whilst in the decoding stage, the ASR output is used for extracting LDA features. This leads to a mismatch between training and testing. The acoustic LDA features that are described next can help address both of these issues.

3) Acoustic-based LDA Auxiliary Features: The main difficulty in extracting LDA features from the acoustic data is that LDA is a model to describe latent factors in sets of discrete symbols [32] which are here interpreted as “domains”, while acoustic features are continuous. In order to fit into that concept speech signals need to be converted into such a form. Typically speech is represented using continuous features such as Mel frequency cepstral coefficients (MFCCs), and has variable length. In our previous work [33] we used the Linde-Buzo-Gray vector quantization algorithm [34] to represent each speech frame with a discrete symbol, equivalent to an acoustic word or phone label.

In this paper an approach similar to that used in [35] has been implemented and utilised. A GMM model with V components is trained using all of the training data. The model is then used to get the posterior probabilities of the Gaussian components to represent each frame with index of the Gaussian component with the highest posterior probability. Frames of every speech segment of length T, $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_t, ..., \mathbf{x}_T\}$ are represented as:

$$\tilde{x}_t = \underset{i}{argmax}\ P(G_i|\mathbf{x}_t) \tag{1}$$

where G_i $(1 \leq i \leq V)$ is the ith Gaussian component. After applying this process to each utterance, each speech segment is represented as $\{\tilde{x}_1, ..., \tilde{x}_t, ..., \tilde{x}_T\}$ where x_t is index of the Gaussian component and thus a natural number $(1 \leq x_t \leq V)$. Here we refer to each speech utterance as an acoustic document.

With this information, a fixed length vector $\hat{\mathbf{x}} = \{a_1, ..., a_i, ..., a_V\}$ of size V is constructed to represent the count of every Gaussian component in an acoustic document. This leads to a bag-of-sounds representation. The sounds would normally be expected to relate to phones, however given the acoustic diversity of background conditions many other factors may play a role. Once these bag-of-sounds representations of acoustic documents are derived, LDA models can be trained. After training the LDA acoustic model, a similar procedure is followed to extract acoustic LDA features from test data.

B. Model-Based RNNLM Adaptation

The most common model-based adaptation of RNNLMs used is fine-tuning, where the RNNLM is further trained on domain-specific text data. In addition to fine-tuning, this work proposes the introduction of a linear hidden network (LHN) adaptation layer to the RNNLM, for domain adaptation. In addition, a modified version of a previously proposed K-component adaptive RNNLM adaptation method [36] is proposed and evaluated on multi-genre speech recognition.

1) Model Fine-tuning: Model fine-tuning involves further training the RNNLM on genre-specific data, thus yielding a separate model per genre. These models can then be used at test time to evaluate text where genre labels are known a-priori, which is the case for MGB data.

2) LHN Adaptation Layer: A hidden layer can be cascaded in the network at adaptation time and only the weights connecting the adaptation layer and the next layer are updated at the time of fine-tuning. Park *et al.*[10] proposed a similar approach for NNLM, by cascading the adaptation layer between the projection and hidden layers. For RNNLMs, a projection layer is not needed and thus, the adaptation layer is cascaded between the hidden and output layers as shown in Figure 2. The adaptation layer has a linear activation and provides a linear transform to the hidden layer and is thus equivalent to the linear hidden network (LHN) [30] transform used in DNN acoustic models [37]. The weights connecting the hidden and adaptation layers are initialised as the identity matrix. This provides an equivalent network to the unadapted RNNLM. At the time of fine-tuning, the rest of the network is kept fixed and only the weights connecting the hidden and adaptation layers are updated.

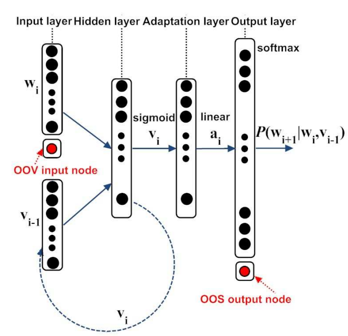


Fig. 2: *RNNLM with LHN adaptation layer.*

3) K-Component Adaptive RNNLMs: The K-Component Adaptive RNNLM was first proposed in [25] for n-gram LMs and in [36] for RNNLMs, where language model training text data was clustered into K clusters, where K is the number of topics in a LDA model [36]. The LDA posteriors were used to partition the text data into clusters by assigning the topic with the highest posterior probability to each sentence in the language model corpus. In [36], K separate topic-specific RNNLMs, $\{M_k\}_{k=1}^{K}$, were obtained by training models on the text derived from clustering. In this work, the topic-based RNNLMs are derived by fine-tuning baseline RNNLMs on the topic-specific text. As a result, the K-Component Adaptive RNNLM described in this paper is a special version of model-based adaptation.

At test time, the K topic-dependent RNNLMs are used in an n-best rescoring framework by interpolating a background language model with the topic-dependent RNNLMs and computing the interpolated probabilities at the sentence level. The sentence probability is computed by linearly combining all the k-component models to assign the sentence probability using the following equation:

$$p(s) = \prod_i \sum_k \lambda_k p(w_i|h_i, M_k) \tag{2}$$

where λ_k is the interpolation weight of the component model k and can be obtained by perplexity-based optimisation on held-out data, w_i is the current word and h_i the history of words, M_K is the k^{th} component model.

In this work, the interpolation weights are taken to be LDA posteriors on the ASR output obtained from a baseline system in first-pass decoding, which gives a more representative distribution similar to that which was used to train the models.

C. Hybrid RNNLM Adaptation

Two hybrid RNNLM adaptation approaches are proposed and investigated in this paper. Both approaches are aimed at leveraging on show-based topic and genre information. The first approach involves fine-tuning LDA feature-adapted RNNLMs on genre-specific text data. The second approach involves introducing a domain-specific genre adaptation layer to a LDA feature-adapted RNNLM.

1) Fine-tuning Feature-Based RNNLM: One way of using topic and genre information effectively for RNNLM adaptation is by fine-tuning LDA-adapted feature-based RNNLMs to genre-specific text, thus also leveraging on the strengths of model and feature-based adaptation.

2) Feature-based RNNLM with Adaptation Layer: The LHN adaptation layer fine-tuning can lead to overfitting if the amount of genre-specific data is not adequate with respect to the model size. Tilk *et al.*[11] showed that the adaptation layer can be made to provide an additive rather than a multiplicative transform (as in the case of the LHN transform). This was achieved by inputting a domain vector d in the form of a 1-of-K encoding, to the adaptation layer. This is equivalent to using genre 1-hot vectors as input to the adaptation layer. Auxiliary features such as LDA, can also be input to the hidden layer as before and this configuration is shown in Figure 3. The advantage of using such a configuration is that shared information between genres is modelled at the adaptation layer and a single model is needed for decoding, as compared to having separate fine-tuned models for each genre.

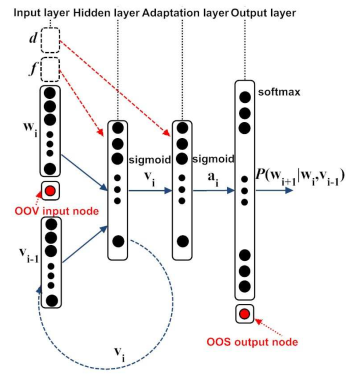


Fig. 3: *RNNLM with feature-based adaptation layer.*

D. Semi-supervised RNNLM Adaptation

In the MGB challenge data, genre information is only available for the transcripts of the acoustic data ($LM2$) and not for the larger (650M words) language model ($LM1$) subtitle text. Genre labels thus need to be automatically derived if we want to apply genre adaptation when using $LM1$ text. It was found that good genre classification can be achieved using support vector machines (SVM), using the LDA features as input, on the $LM2$ text.

V. RNNLM Adaptation for ASR

The stages involved in using RNNLM adaptation for ASR are as follows. Voice Activity Detection (VAD) is first applied to the audio in order to identify speech segment boundaries. The input text is then converted to a monophone/triphone/senome representation and aligned to the segmented audio using a baseline ASR system. The segmented audio and aligned text are fed to a DNN-HMM system which can be either a Hybrid or Bottleneck system [38]. In the Hybrid system, a DNN is used to predict monophone/triphone/senome states from audio features, which in most cases are log Filterbank features. This results in posteriors over these states, which are integrated as observation probabilities in a hidden Markov model (HMM), and used to predict the optimal path by also taking into account dynamical constraints arising from an underlying language model. In a Bottleneck system, the log Filterbank features are fed to the DNN as input and the monophone/triphone/senome states as output. A Bottleneck layer is introduced between the final layer of the DNN and the output layer which generally has a lower dimension than the final layer. The activation values of that layer are then extracted as Bottleneck features. These Bottleneck features are used as input to a standard GMM-HMM system and have been found to outperform GMM-HMM systems with MFCC or PLP features [38], due to the discriminative nature of the input features.

At decoding stage, input audio is given and segmented using the VAD system similarly as for training. This allows log Filterbank features to be extracted from the audio segments. These features are fed to the DNN-HMM system together with a baseline n-gram, out of which a hypothesis ASR output is generated either in the form of lattices or n-best lists. These lattices or n-best lists can then be rescored using an adapted

RNNLM where the language model scores are interpolated between the n-gram scores and the RNNLM scores and a 1-best output is produced.

Whilst segmentation using VAD was performed in our systems for the MGB challenge [14], for the purpose of this paper, manual segmentations provided for the training and development sets are used, as this factors out any errors that are introduced from automatic segmentation.

VI. RNNLM Adaptation for LSA

In LSA [39], [40], [41], [42], the input audio is first processed by VAD. Meanwhile, the input subtitles are also processed using standard text normalisation and tokenisation. This is followed by training a biased language model adapted to these subtitles. For n-gram LMs, this can be achieved by interpolating the larger background language model with a subtitle language model [41]. The adapted language model is then used in the decoding of the input audio. This stage is usually referred to as lightly supervised decoding [39], since it uses a language model biased to the subtitles. Multiple decoding passes and speaker adaptation can be carried out as necessary.

In this work RNNLM adaptation is used as part of the lightly supervised decoding process, in order to get an output that most closely matches the ground truth. The lightly supervised decoding is thus performed in 3 stages. First decoding is performed using a set of previously trained hybrid DNN-GMM-HMM acoustic models used for ASR and a baseline 3-gram language model trained on $LM1\&LM2$ text. This generates a set of lattices which are then rescored using the biased n-gram language model. The rescoring process generates a set of n-best lists, which can then be further rescored using adapted RNNLMs. The RNNLMs need to be biased to each show and this can be performed using show-based fine-tuning. In this work, the show subtitle text is used both as the training and development sets and using 10 epochs of further training in order to prevent overfitting. The same learning rate used to train the model is also used for fine-tuning. This step was found to be crucial for biasing the RNNLM whilst setting the learning rate to the level reached at the last epoch led to poor convergence.

An alignment stage then aligns the transcript hypothesis given by the lightly supervised decoding stage, to the original input text. Dynamic programming approaches can be used for this, where sequences of words from the subtitles are assigned to the speech segments, based on how well they match the output of the lightly supervised decoding, using a distance metric such as Levenshtein distance [40]. The lightly supervised alignment stage results in a set of speech segments whose transcripts contain words from the original subtitle text. This is then followed by a second alignment stage to provide precise word-boundaries for the output, thus completing the process.

VII. Experiments and Results

For both ASR and LSA experiments, the MGB development data was used as the test set, so as to allow comparison with previous work [6], [43] and also because it has been shown that the ASR results on the development and evaluation sets are highly correlated [14]. A partitioning of the $LM2$ text into a training and development set was carried out by first shuffling the lines for each programme and selecting 90% of text for each programme for training and the remaining 10% for development.

A. ASR Results

1) Experimental Setup: For training the baseline 4-gram LM, a vocabulary of 200k words was built. This was taken from all the words in the $LM2$ text (32k) and topping it up with the most frequently occuring words in $LM1$. The aim of this procedure is to use all words in the in-domain set and increase the vocabulary to include words in the out-of-domain set, so as to provide wider coverage. For acoustic modelling, we selected 700h. of speech from the training set based on word matching error rate (WMER) and confidence scores [44]. The acoustic models used in this work were Bottleneck DNN-GMM-HMM trained using TNet [45] and HTK [46] toolkits, where TNet was used for extracting 26 bottleneck features using a DNN. The DNN 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. It took as input 15 contiguous log-filterbank frames. Discriminative training in the form of state-level Minimum Bayes Risk (sMBR) [47], [48] was used as the DNN target function. 65-dimensional feature vectors were used for training the GMM-HMM systems, which included the 26 dimensional Bottleneck features, as well as 13 dimensional PLP features together augmented with first and second derivatives. 16 Gaussian components per state were used for training the GMM-HMM models, which had about 8k distinct triphone states.

The SRILM toolkit [49] was used for training our baseline 4-gram language model on $LM1\&LM2$ text, using the 200k vocabulary. Kneser-Ney smoothing is used for building the 4-gram LM using the corresponding options provided in SRILM. The RNNLMs were trained used a modified version of the CUED-RNNLM toolkit [50]. For training the baseline RNNLM, we used the full $LM1\&LM2$ text, together 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 frequency-based shortlisting of the 200k vocabulary and the main reason is to reduce computational complexity in computing of the RNNLM output softmax layer. The OOV and OOS nodes deal with words in the original vocabulary but not in the shortlisted $60k$ and $50k$ vocabularies respectively. The RNNLM training regime used was with mini-batch of 128 and learning rate of 2.0, which resulted in an effective learning rate of 0.0156 per mini-batch. The learning rate was halved for each subsequent epoch if the change in entropy on the development data from the previous epoch to the current is less than 1% and model training stopped upon convergence.

A 3-stage process was following for decoding with Bottleneck systems. In the first stage, lattices were generated using a 2-gram LM. This was followed by lattice rescoring with the 4-gram LM to generate new lattices. Further lattice rescoring of

TABLE IV: *RNNLM baseline ASR results.*

	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

TABLE V: *RNNLM adaptation results using only in-domain ($LM2$) RNNLM.*

	Genre →	Adv.	Child.	Comed.	Compet.	Docum.	Dram.	Even.	News	Global	
System	Adaptation	WER								PPL	WER
$LM1\&LM2$ 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
K-Component Adapative Topic fine-tuning using LDA posteriors	Model	24.3	30.0	43.6	25.5	27.9	41.7	33.4	15.3	93.8	29.9
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
Text 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
Acoustic LDA feat. at hidden layer	Feature	24.3	29.5	43.5	25.4	27.6	41.8	33.1	14.9	92.9	29.7
Acoustic+Text LDA feat. at hidden layer	Feature	24.0	29.5	43.6	25.5	27.5	41.7	33.1	15.0	90.4	29.6
Text 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
Text LDA feat. at hidden layer and no feat. at adapt. layer	Hybrid	23.7	29.2	43.3	25.1	27.3	41.6	32.8	14.7	87.5	29.4
Text LDA feat. at hidden layer, no feat. at adapt. layer and genre fine-tuning	Hybrid	23.7	29.0	43.2	24.8	27.2	41.7	32.7	14.6	87.1	29.3
Text LDA feat. at hidden layer and genre feat. at adapt. layer	Hybrid	23.6	28.9	43.4	24.9	27.3	41.2	32.5	14.6	86.9	29.2

TABLE VI: *RNNLM adaptation results using in-domain and out-of-domain ($LM1\&LM2$) RNNLM.*

	Genre →	Adv.	Child.	Comed.	Compet.	Docum.	Dram.	Even.	News	Global	
System	Adaptation	WER								PPL	WER
$LM1\&LM2$ 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
K-Component Adapative Topic fine-tuning using LDA posteriors	Model	23.5	28.9	42.8	24.6	26.8	41.2	32.4	14.1	81.8	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
Text 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
Text 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
Text LDA feat. at hidden layer and no feat. at adapt. layer	Hybrid	23.0	28.5	42.7	24.5	26.5	40.5	32.5	14.3	80.9	28.7
Text LDA feat. at hidden layer, no feat. at adapt. layer and genre fine-tuning	Hybrid	22.9	28.6	42.5	24.2	26.4	40.3	32.3	14.5	79.9	28.7
Text LDA feat. at hidden layer and genre feat. at adapt. layer	Hybrid	22.9	28.6	42.5	24.2	26.4	40.3	32.3	14.1	79.4	28.6

the 4-gram lattices was then performed using the RNNLM, according to the nth-order truncation method described in [51], with n being set to 6, in line with what was reported in [51]. We compared this against n-best list rescoring by first converting the lattices to n-best lists, followed by 1-best computation. n was set to 100 after varying n from 100 to 1000 in increments of 100 and computing 1-best WER on the development set. It was found that for n higher than 100, there is only marginal improvements in the WER and thus n was fixed to 100.

The baseline results, used to investigate the performance of lattice-rescoring vs. n-best list scoring are shown in Table IV. The adaptation results using $LM2$ RNNLM are given in Table V whilst adaptation results using $LM1 + LM2$ RNNLM are given in Table VI. All the results were obtained by scoring using the official MGB scoring package [14], which gives word error rate decompositions across the 8 genres. Perplexities (PPL) are also reported on the reference text. 512 nodes were used for both the hidden layer and adaptation layers as this was found to give good performance whilst remaining computationally efficient. For $LM1\&LM2$ RNNLMs, it was found that n-best list rescoring gives an improvement of 0.2% in the WER over lattice rescoring. As a result, we used n-best list rescoring for all RNNLM adaptation experiments. The interpolation weight with the 4-gram baseline LM was set to 0.5 and 0.3 respectively for $LM1\&LM2$ and $LM2$ RNNLMs, as this was found to give the lowest PPL on our development set.

2) Feature-based Adaptation Results: We found text LDA auxiliary features to be more effective than genre 1-hot features for the adaptation of $LM2$ RNNLMs, similar to what has been reported in the literature [6], [43]. In order to choose the LDA feature dimensionality, we extracted LDA features from the reference text for each show and varied the number of topics from 10 to 150 and computed the PPL on our development set. 100 topics was found to give the best result and as a result, the number of LDA topics was fixed to 100. It is to be noted that larger values for n result only in marginal gains in PPL, at the expense of increased computational cost. Chen *et al.*[6] showed that by using 30 LDA topics and computing the LDA features from the ASR output instead of the reference text, a degration in WER of about 0.1% was observed. In our case, it was found that with 100 topics, extracting LDA features on both reference and ASR output, lead to the same overall WER result, albeit some minor variations within genres. The ASR output text from first-pass decoding was thus used to compute LDA features, as would be expected in a practical system. An improvement in WER of 0.3% is obtained when using text LDA features over the RNNLM baseline. For acoustic LDA, we similarly varied the number of LDA topics but used exponents of 2 this time, in line with what was presented in [33]. 128 topics was found to give the best results in terms of PPL. Acoustic LDA features were found to give comparable performance to genre 1-hot features with a WER of 29.7%. This shows that acoustic LDA features do not provide as good a representation of domain as text LDA features. However, they can be used in single-pass decoding as opposed to two-pass decoding in the case of text LDA features. Combining text and acoustic LDA features gives a WER of 29.6%, which is slightly worse than when using only text LDA features (29.5% WER). As a result, it can be inferred that text LDA features provide a much better representation of domain than acoustic LDA features and the latter does not complement the former.

LDA feature-based adaption of $LM1\&LM2$ RNNLM with 100 topics, was also found to result in a substantial drop in WER of 0.5%. Moreover, the genre fine-tuning of LDA-adapted RNNLMs lead to small but significant gains in WER. For the $LM2$ RNNLM, this hybrid adaptation leads to a global drop in WER from 29.5% to 29.4%.

3) Model-based Adaptation Results: The experiments show that LHN adaptation layer fine-tuning of RNNLMs, outperforms full model fine-tuning in terms of PPL, for both $LM2$ and $LM1\&LM2$ RNNLMs. For the RNNLM trained on $LM1\&LM2$ text, there is also a drop of 0.1% in the global WER (28.9%) using the LHN adaptation layer, compared to full model fine-tuning (29.0%). For K-component topic-based RNNLM adaptation, K was fixed to 10 as this was found to give optimal performance when K was varied from 2 to 20 in increments of 2. It was found that the K-component topic-based RNNLM adaptation performs better when using $LM1\&LM2$ RNNLM with a WER of 28.9% and better than the result obtained using genre fine-tuning. This shows that topic-based fine-tuning together with an interpolation of topic-dependent RNNLMs based on LDA posteriors gives better results than when using genre fine-tuned RNNLMs, where a hard decision is made about the genre-specific model to be used for n-best list rescoring for each utterance in the test set. On the other hand, the result obtained using K-Component topic-based fine-tuning of $LM2$ RNNLM (WER 29.9%) is slightly worse than the result obtained with genre fine-tuning (WER 29.7%), thus pointing out to the technique performing less well when using an RNNLM trained on smaller amounts of data. This was found to be due to the adapted model over-fitting to the topic-dependent data and loosing its generalisation ability, when using a smaller unadapted model ($LM2$) compared to one trained on a larger set ($LM1\&LM2$). The non-interpolated training PPL for $LM2$ RNNLM was found to be 101.4 using the K-component topic-based fine-tuning compared to 106.7 for using text LDA adaptation and 119.2 when not using adaptation (baseline), whereas the non-interpolated test PPL was: 157.2, 136.4 and 150.8 respectively, which shows overfitting of the K-Component RNNLM. Our results with the introduction of a domain-specific adaptation layer showed that using an adaptation layer with additive bias adaptation (feature-based adaptation layer), better results are obtained than when using a multiplicative transform (LHN adaptation layer), which is in line with similar observations in acoustic modelling [52]. For $LM1\&LM2$ RNNLMs, an additive transform gives a WER of 28.7% whereas using a multiplicative transform leads to a WER of 28.9%. This improvement is however, not observed with $LM2$ RNNLM, with both additive and multiplicative transforms giving comparable performance. One possibility for this could be that a multiplicative transform is more prone to over-fitting, especially when the amount of in-domain data is limited, as is the case for genres such as comedy and drama. In order to allow for a fair comparison, baselines have been added with 2-layer RNNLMs (no feat. at adaptation layer) which show that adding an extra feed-forward layer with no adaptation leads to improvements in the results as expected, but adding the genre features at the adaptation layer lead to even further improvements.

4) Semi-supervised and Hybrid Adaptation Results: When using the SVM to predict genre labels on the official MGB development set, it was found that 1024 topics gave the best classification accuracy of 94.79%. We thus use this model to predict the genre labels for $LM1$ text in order to provide the genre feature input for $LM1\&LM2$ RNNLMs.

It is shown by the results that genre labels obtained using SVM classification on $LM1$ text, together with the $LM2$ text genre labels, can be used as input features to the adaptation layer for the $LM1\&LM2$ RNNLM, leading to a drop in WER of 0.5% from 29.2% to 28.7%. Moreover, it is also found that for $LM1\&LM2$ RNNLMs trained with those LDA-derived genre labels at the adaptation layer, comparable results are obtained to using LDA features input to the hidden layer with a WER of 28.7% for both, although combining the two inputs yields a further improvement to 28.6%.

The results show that the two domain representations, i.e. LDA topic features and genre labels provided as part of the MGB challenge. Combining 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 a $LM2$ RNNLM and from 29.2% to 28.6% using a $LM1\&LM2$ RNNLM. It should be noted that all the $LM2$ and $LM1\&LM2$ RNNLM adaptation results are statistically significant ($p < 0.05$) with respect to the no adaptation baseline, using the Matched Pair Sentence Segment, Signed Paired Comparison, Wilcoxon Signed Rank and McNemar tests done with `sc_stats`, provided as part of the NIST SCTK package[1].

B. Alignment Results

1) Experimental Setup: The lightly supervised decoding stage operated as follows: first, a DNN–based speech segmentation module is used to identify segments of speech in the show. An initial transcription for these segments is obtained from a speaker independent DNN–HMM system [38] trained on 700 hours of acoustic training data using the Kaldi toolkit [53]. This stage also uses the background 4-gram language

[1]https://www.nist.gov/itl/iad/mig/tools

TABLE VII: *Lightly supervised alignment results.*

Language Model Configuration	PPL	WER	Precision	Recall	F–measure
Show adapted N-gram	44.7	23.6	85.40	88.40	86.88
+Show adapted RNNLM	35.0	22.3	85.16	88.63	86.93
+Text (subtitle) LDA feat. at hidden layer	29.5	21.9	85.31	89.62	87.41
+Text (subtitle) LDA feat. at hidden layer and Genre feat. at adaptation layer	26.6	21.8	85.10	89.68	87.33

model trained on the subtitle data using SRILM [54]. This output is then used for re–segmentation, speaker clustering and speaker adaptation to the second decoding stage, based on a DNN–GMM–HMM system trained on 700 hours of speech. This second stage uses an interpolation of a n-gram language model, trained on $LM1\&LM2$ text and biased towards the subtitles as described in section VI, and a biased RNNLM involving the fine-tuning of a baseline RNNLM to show-specific text, with an interpolation weight of 0.5. Finally, the output is aligned to the subtitles in a recursive lightly supervised alignment stage. Boundary correction techniques presented in [55] can further be applied as a post-processing step, but they were not considered in this work.

Instead of evaluating all the RNNLM adaptation combinations from the ASR experiments, the models that gave the best results are evaluated for aligment. In particular, RNNLM adaptation using LDA features derived from the subtitles as input to the hidden layer, and the hybrid adaptation approach using the combination of LDA features at the hidden layer and genre 1-hot features at the adaptation layer, are evaluated.

The alignment results are reported as both the perplexity on the reference show text and word error rate obtained by scoring the ASR output obtained after decoding/rescoring using biased language models against the reference, using the official MGB ASR scoring package [14], and as precision, recall and F–measure obtained using the official MGB alignment scoring package [14] with further details given in [56]. Alignment is framed as a word detection task, where words are individually assessed to be correctly or incorrectly aligned. The precision, recall and F–measure are computed as follows:

$$Precision = \frac{N_{match}}{N_{hyp}} \tag{3}$$

$$Recall = \frac{N_{match}}{N_{ref}} \tag{4}$$

$$F = 2\frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

where N_{hyp} is the number of words in the hypothesis/ASR output that are also in the subtitles to be aligned, N_{ref} is the number of words in the reference that are also in the subtitles. N_{match} is the number of matching word counts.

The results in Table VII show that the use of a show-adapted RNNLM decreases the WER from 23.6% to 22.3%, resulting in an increase in F–measure from 86.88% to 86.93%. Using LDA features as input to the hidden layer results in a further reduction of the WER to 21.9% and an increase in the F–measure to 87.41%. Using LDA features at the hidden layer and genre features at the adaptation layer leads to a further reduction of the WER to 21.8% but F–measure drops slightly 87.33% with a small decrease in precision but a small improvement in recall. The WER results are consistent with what was obtained in ASR experiments and show text LDA features derived from subtitles to be effective for RNNLM adaptation when using show-based fine-tuning for biasing the RNNLM to each show. Using a combination of LDA and genre information did not lead to an improvement in F–measure but leads to a different distribution of the results in terms of precision and recall.

VIII. Conclusion

In this work, various feature and model-based adaptation methods for RNNLMs have been compared and combined on multi-genre speech recognition and alignment. The two approaches were found to be complementary and a combination of both types of adaptation generally improves performance. For feature-based adaptation, the use of latent Dirichlet allocation (LDA) features as augmented feature input to the RNNLM was found to be effective for both ASR and alignment, leading to a reduction in WER of 0.5% for ASR and an increase in F–measure of about 0.5% for alignment. The use of acoustic LDA features have also been investigated for ASR and it was found that text LDA features are more effective than acoustic LDA features. However, it was demonstrated that acoustic LDA features can be useful in a first-pass ASR system as the text LDA features need to be extracted from a first-pass ASR system and used for RNNLM rescoring of lattices or n-best lists in the second pass. Model-based adaptation techniques that have been investigated in this paper include: the use of a feature-based adaptation layer which provide an additive bias adaptation to specific genres; the use of linear hidden network (LHN) adaptation layer for genre-based fine-tuning; full model genre-based fine-tuning and a K-component RNNLM adaptation using topic-based fine-tuning, where the topic posteriors are derived using LDA. We proposed two hybrid adaptation techniques that harness on the strengths of feature and model-based adaptation, both of which leveraging on the combination of topic and genre information for optimal performance.

One limitation of the proposed hybrid methods is that it assumes that the genres are known at training time. Unseen genres can be dealt with by using a place-holder at training time (e.g. a zero vector) and then introducing the genre auxiliary code at fine-tuning once it becomes known. This has been investigated in a recent paper by the same authors [57]. This paper is based on a previous Interspeech paper [31] which included the bulk of the results in Tables IV, V and VI but with additional experiments and more detailed investigations, as well as investigating the proposed methods for lightly supervised alignment.

Acknowledgment

The audio and subtitle data was made available as part of the MGB Challenge (mgb-challenge.org) through a licence with the BBC. The CTM and scoring files for the

ASR experiments in the paper can be accessed via DOI: 10.15131/shef.data.3581427. This work was supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] S. Reddy, P. Swietojanski, P. Bell, and S. Renals, "Unsupervised adaptation of recurrent neural network language models," in *INTERSPEECH'16: Proc. of the Annual Conference of the International Speech Communication Association*, 2016, pp. 2333–2337.

[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. J. Moreno, C. F. Joerg, J. V. Thong, and O. Glickman, "A recursive algorithm for the forced alignment of very long audio segments," in *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference*, 1998.

[13] N. Braunschweiler, M. J. F. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *INTERSPEECH'10: Proc. of the 11th Annual Conference of the International Speech Communication Association 2010*, 2010, pp. 2222–2225.

[14] 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.

[15] 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.

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

[17] 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.

[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] 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.

[20] 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.

[21] M. Sundermeyer, I. Oparin, J. Gauvain, B. Freiberg, R. Schlüter, and H. Ney, "Comparison of feedforward and recurrent neural network language models," in *ICASSP'13: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8430–8434.

[22] J. R. Bellegarda, "Statistical language model adaptation: review and perspectives," *Speech Communication*, vol. 42, no. 1, pp. 93 – 108, 2004.

[23] X. Liu, M. J. F. Gales, and P. C. Woodland, "Context dependent language model adaptation," in *Interspeech'08: Proc. of the Annual Conference of the International Speech Communication Association*, 2008.

[24] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech & Language*, vol. 10, no. 3, pp. 187 – 228, 1996.

[25] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *ICASSP'03: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2003, pp. 224–227.

[26] A. Ter-Sarkisov, H. Schwenk, F. Bougares, and L. Barrault, "Incremental adaptation strategies for neural network language models," in *CVSC'15: Proc. Workshop on Continuous Vector Space Models and their Compositionality*, 2015.

[27] M. Ma, M. Nirschl, F. Biadsy, and S. Kumar, "Approaches for neural-network language model adaptation," in *Interspeech 2017: 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 259–263.

[28] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, 2009.

[29] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep neural network language models," in *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, ser. WLM '12. Association for Computational Linguistics, 2012, pp. 20–28.

[30] 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.

[31] 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 *Proceedings of the 17th Annual Conference of the International Speech Communication Association (Interspeech)*, San Francisco, California, USA, 2016.

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

[33] M. Doulaty, O. Saz, and T. Hain, "Unsupervised domain discovery using latent Dirichlet allocation for acoustic modelling in speech recognition," in *Proc. of Interspeech*, Dresden, Germany, 2015.

[34] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Berlin, Germany: Springer Science & Business Media, 1992.

[35] C. Ni, C. C. Leung, L. Wang, N. F. Chen, and B. Ma, "Unsupervised data selection and word–morph mixed language model for tamil low-resource keyword search," in *Proc. of ICASSP*, Brisbane, Australia, 2015.

[36] Y. Shi, M. Larson, P. Wiggers, and C. M. Jonker, "K-component adaptive recurrent neural network language models," in *Proc. of 16th International Conference on Text, Speech, and Dialogue*, 2013, pp. 311–318.

[37] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *INTERSPEECH'10, Proc. of the 11th Annual Conference of the International Speech Communication Association, 2010*, 2010, pp. 526–529.

[38] G. Hinton, L. Deng, D. Yu, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, G. Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[39] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[40] A. Katsamanis, M. Black, P. Georgiou, L. Goldstein, and S. Narayanan, "SailAlign: Robust long speech–text alignment," in *Proceedings of the Workshop on New Tools and Methods for Very Large Scale Research in Phonetic Sciences (VLSP)*, Philadelphia, PA, 2011, pp. 44–47.

[41] P. Lanchantin, M. J. Gales, P. Karanasou, X. Liu, Y. Qian, L. Wang, P. C. Woodland, and C. Zhang, "The Development of the Cambridge University Alignment Systems for the Multi-Genre Broadcast Challenge," Scottsdale, AZ, 2015, pp. 647–654.

[42] A. Stan, Y. Mamiya, J. Yamagishi, P. Bell, O. Watts, R. Clark, and S. King, "Alisa: An automatic lightly supervised speech segmentation and alignment tool," *Computer, Speech & Language*, vol. 35, pp. 116–133.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

[50] X. Chen, X. Liu, Y. Qian, M. J. F. Gales, and P. C. Woodland, "CUED-RNNLM - an open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *ICASSP'16: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 6000–6004.

[51] 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.

[52] 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.

[53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," 2011.

[54] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, Denver, CO, 2002, pp. 901–904.

[55] J. Olcoz, O. Saz, and T. Hain, "Error correction in lightly supervised alignment of broadcast subtitles," in *Proceedings of the 17th Annual Conference of the International Speech Communication Association (Interspeech)*, San Francisco, CA, 2016.

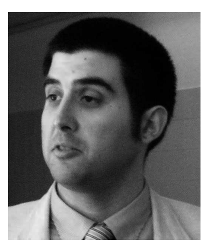
[56] T. Hain, "MGB Challenge 2015 Task 2 Alignment of broadcast audio to a subtitle file," Natural Speech Technology, Tech. Rep., 2015. [Online]. Available: http://data.cstr.ed.ac.uk/asru/scoring/eval_task2.pdf

[57] S. Deena, R. W. M. Ng, P. Madhyashta, L. Specia, and T. Hain, "Semi-supervised Adaptation of RNNLMs by Fine-tuning with Domain-specific Auxiliary Features," in *INTERSPEECH'17: Proc. of the Conference of the International Speech Communication Association*. ISCA, 2017.

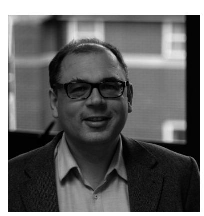
Salil Deena is a Research Associate in the Speech and Hearing (SpandH) group at the University of Sheffield where he has been a member of the Natural Speech Technology (NST) project. He received his B.Sc. (Hons) in Information Technology from the University of Mauritius in 2005, his M.Sc. and Ph.D in Computer Science from the University of Manchester in 2007 and 2012 respectively. His research interests are in Deep Learning approaches to natural language processing, speech technology and computer vision.

Madina Hasan received her B.Sc. in Electronics Engineering from The University of Bahrain in 2004, her M.Sc. in Data Communications and Ph.D in Computer Science from The University of Sheffield, UK in 2007 and 2012 respectively. Since 2012, she is a research associate at the University of Sheffield. Her research interests are in applying machine learning techniques for natural language processing problems.

Mortaza Doulaty received his B.Sc. (Hons) and M.Sc. in Computer Science from the University of Tabriz, Iran in 2009 and 2011. He recently received his Ph.D. candidate in Computer Science at the Machine Intelligence for Natural Interfaces (MINI), Speech and Hearing Group (SPandH), University of Sheffield, UK. His research interests are domain discovery and domain adaptation in speech recognition.



Oscar Saz received his B.Sc. in Telecommunications Engineering from the University of Zaragoza, Spain in 2004 and his Ph.D. in 2009 from the same institution. From 2010 to 2012 he was a Fulbright scholar at Carnegie Mellon University and has been a Research Associate at the University of Sheffield until 2016. His interests are acoustic modelling and adaptation for Automatic Speech Recognition.



Thomas Hain is Professor for Computer Science at the Unversity of sheffield. He holds the degree 'Dipl.-Ing' in Electrical and Communication Engineering from the University of Technology, Vienna, and a PhD in Information Engineering from Cambridge University (2002). After work at Philips Speech Processing, Vienna he joined the Cambridge University Engineering Department in 1997, where he was appointed to Lecturer in 2001. He then moved to Sheffield in 2004, and was promoted to Professor in 2012. Prof Hain leads the 15-strong subgroup on Machine Intelligence for Natural Interfaces and has more than 140 publications on machine learning and speech recognition topics.