# Unsupervised Language Model Adaptation for Mandarin Broadcast Conversation Transcription

David Mrva and Philip C. Woodland

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, UK
{dm312,pcw}@eng.cam.ac.uk

## Abstract

This paper investigates unsupervised language model adaptation on a new task of Mandarin broadcast conversation transcription. It was found that N-gram adaptation yields 1.1% absolute character error rate gain and continuous space language model adaptation done with PLSA and LDA brings 1.3% absolute gain. Moreover, using broadcast news language model alone trained on large data under-performs a model that includes additional small amount of broadcast conversations by 1.8% absolute character error rate. Although, broadcast news and broadcast conversation tasks are related, this result shows their large mismatch. In addition, it was found that it is possible to do a reliable detection of broadcast news and broadcast conversation data with the N-gram adaptation.

**Index Terms**: speech recognition, language model adaptation, Mandarin, broadcast conversation transcription

## 1. Introduction

This paper investigates language model adaptation in a speech recognition setting with a mix of data of two mismatching styles. Several language model adaptation techniques were examined on a recently introduced broadcast conversation (BC) transcription task. All experiments were carried out on Mandarin. Broadcast conversations refer to free speech that occurs in news-style TV and radio shows; i.e. interviews, debates, listeners/viewers calling in, live reports.

BC transcription task is closely linked to the transcription of broadcast news (BN). The main challenge is both BC and BN data being mixed together without an indication of which piece of the input is BC and which piece is BN data. The aim is to construct a transcription system that will take recordings from TV or radio that may contain both BN and BC-style speech and transcribe these recordings without a manual classification.

The crucial difference between BC and BN is the large mismatch in style: BN is read prepared speech whereas BC is spontaneous speech. Due to this mismatch, a system trained on BN data only does not give good results on BC data. As the differences come from the style of speech, LM should detect BN and BC portions of the input. A method for this detection is described in this paper.

The first problem to tackle was how to split the input into the pieces to be classified. To keep the system simple, the inherent structure of the data was exploited. All the test sets comprise of shows where a show is a broadcast of a TV or radio program. The two simplest levels for classification available are shows and segments. Although segments are far more homogeneous than shows, one segment is very short, therefore, whole shows were used for the BC/BN classification.

The next important problem was what method to use to detect the BC/BN data. As shows are typically sufficiently large, it was expected that optimising the interpolation weights of BN and BC N-grams on individual shows would provide a good guidance for the BC/BN detection. The empirical results in this paper showed that it is possible to do this detection based on the show-specific interpolation weights.

The purpose of the classification is for a speech recogniser to change some of its parameters for different portions of the input. Given the nature of the mismatch, an obvious place for such changes is the language model. There is a well-trained BN LM available for the BN-style speech. However, there is not enough data to train a good BC language model. The usage of an interpolated N-gram model with dynamic interpolation weights and two continuous space adaptive language models were tested. The next section describes the tested models, followed with experimental results.

## 2. Methods for language model adaptation

### 2.1. Adaptation of N-gram mixture model

An N-gram mixture model was used as a baseline throughout this paper. To obtain an adapted N-gram model, the interpolation weights of this baseline model were optimised with standard EM algorithm on a hypothesis from a previous pass of a multi-pass system. The mixture model had the form of

$$\begin{aligned} P(w_t|w_{t-3}^{t-1}) &= \lambda P_{\text{BCM}}(w_t|w_{t-3}^{t-1}) + \\ &\quad (1-\lambda) P_{\text{BNM+ENG}}(w_t|w_{t-3}^{t-1}) \end{aligned}$$

where $w_{t-3}^{t-1} = w_{t-3}, w_{t-2}, w_{t-1}$, $\lambda \geq 0$, $P_{\text{BCM}}$ is a four-gram model trained on BC data and $P_{\text{BNM+ENG}}$ a four-gram model trained on BN data. The training of these two components will be described in more detail in section 3.2.

### 2.2. Adaptive language models with hidden variables

Due to little BC data being available, it is desirable to use the data as efficiently as possible. Adaptive techniques that take into account word co-occurrences in addition to N-gram statistics were investigated. Models based on word co-occurrences exploit the fact that some words occur often in the same documents. These models make the assumption that if some group of words often appears in the same documents, their probabilities should be increased if one of these words is seen. Two such models were

examined: Probabilistic Latent Semantic Analysis (PLSA)-based LM [3, 4, 7] and Latent Dirichlet Allocation (LDA) LM [1, 6]. Alternatively, these models can be seen as methods that project documents to a continuous vector space and perform the adaptation in this vector space.

A document in these models is a logical piece of text; e.g. an article. In speech recognition, it is a piece of speech; e.g. a conversation side, a broadcast news show etc. In training, every show was split into speaker-specific portions and these portions were used as documents. Each training document contains segments from one speaker and one show only which gives the model a chance to capture patterns in word usage on both speaker and show levels. In testing, whole show was a document.

The PLSA and LDA document probabilities are closely related. For a document $D$ that contains a sequence of words $\mathbf{w} = w_1, \ldots, w_{|D|}$, PLSA defines the probability P(**w**) as a product of mixtures of word distributions $P(w_i|z_k)$ that depend on the value of a hidden variable $z_k$.

$$P(\mathbf{w}) = \prod_{t=1}^{|D|} \sum_{k=1}^{K} P(w_t|z_k)P(z_k|D) \tag{1}$$

where $P(z_k|D)$ are word distributions' weights fixed for a given document, $w_t$ is the $t$-th token, $|D|$ is the number of tokens in $D$, and $K$ is the number of discrete values attainable by the hidden variable.

On the other hand, LDA adds a prior distribution $p(\theta; \alpha)$ to loosen the constraint of document-specific fixed weights $P(z_k|D)$. This prior distribution generates a multinomial distribution $\theta$ and then the weights $P(z_k|\theta)$ are sampled for this $\theta$: $P(z_k = n|\theta) = \theta_n; n = 1, \ldots, K$. LDA integrates over all values of $\theta$ to calculate the document probability.

$$P(\mathbf{w}) = \int_{\theta} \left( \prod_{t=1}^{|D|} \sum_{k=1}^{K} P(w_t|z_k)P(z_k|\theta) \right) p(\theta; \alpha) d\theta \tag{2}$$

where $p(\theta; \alpha)$ is a Dirichlet prior with parameters $\alpha = \langle \alpha_1, \ldots, \alpha_K \rangle$.

The following sections 2.2.1 and 2.2.2 provide a brief summary of PLSA and LDA-based language models.

#### *2.2.1. PLSA-adapted model*

The PLSA model is a unigram model that is used in a PLSA-based LM as a way of boosting the probabilities of some words and decreasing the probabilities of other words. Which probabilities are increased and which are decreased depends on the ratio of the PLSA and standard unigram probabilities. This ratio is used as a multiplicative factor of an N-gram as shown below.

$$P(w_i|h_i) \propto P_{\text{N-gram}}(w_i|h_i) * \frac{P_{\text{PLSA}}(w_i|h_i)}{P_{\text{unigram}}(w_i)} \tag{3}$$

The history $h_i$ is fixed throughout a segment and comprises of all segments in the current show except the current segment. Keeping the history fixed for a segment and therefore ignoring the long-term context within a segment makes the rescoring feasible.

The PLSA probability $P_{\text{PLSA}}(w_i|h_i)$ is defined as a sum over all values of a hidden variable $z_k$:

$$P_{\text{PLSA}}(w_i|h_i) = \sum_{k=1}^{K} P(w_i|z_k)P(z_k|h_i) \tag{4}$$

A value of the hidden variable $z_k$ represents a characteristic of the text determined by word co-occurrences. Both $P(w_i|z_k)$ and $P(z_k|h_i)$ are trained with EM optimising the training data log-likelihood.

The PLSA training procedure requires a collection of documents. The first step is constructing a word-by-document matrix where each element is set to $n(d_i, w_j)$ the number of occurrences of the $j$-th vocabulary entry $w_j$ in a training document $d_i$. The EM procedure starts from a random initialisation of the model parameters $P(w_j|z_k)$ and $P(z_k|d_i)$ iterating the E-step; e.q. (5), and M-step; eqs. (6) and (7) until convergence.
E-step

$$P(z_k|d_i, w_j) = \frac{P(z_k|d_i)P(w_j|z_k)}{\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)} \tag{5}$$

M-step

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)} \tag{6}$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j)P(z_k|d_i, w_j)}{n(d_i)} \tag{7}$$

where $N$ is the number of training documents and $M$ is size of the vocabulary.

The probabilities $P(w_i|z_k)$ are calculated in training and stay fixed in testing. The model parameters that are adapted are aspect mixture weights $P(z_k|h_i)$ that are calculated for each segment.

When using the PLSA model in testing, the first step is to calculate the aspect weights for the history of a given segment. After that, the word probabilities for the words in the current segment are calculated. Then the model moves to the next segment and repeats both steps. Note that in the PLSA model, the PLSA history includes all words from a test document both before and after the current segment.

For a given history, the aspect weights are calculated with incremental EM which iterates over all the words in the history and updates the aspect weights $P(z_k|h_i)$. There are two cases to be distinguished. For the first word of the history, the topic distribution defaults to the distribution observed in the training data.

$$P(z_k|\hat{h_1}) = P(z_k) = \frac{\sum_{w,d} n(w, d)P(z_k|d)}{\sum_{w,d} n(w, d)} \tag{8}$$

where $\hat{h_1}$ is a partial history which is empty for the first word. $n(w, d)$ is the number of occurrences of a word $w$ in a training document $d$ and the sums iterate over all words in the vocabulary and all training documents. For all other words $w_t$ in the history of a given segment, the partial history $h_t$ is $w_1, \ldots, w_{t-1}$ and the incremental EM updates the aspect weights in the following way:

$$P(z_k|h_t) = \frac{1}{t+1} \frac{P(w_t|z_k)P(z_k|h_{t-1})}{\sum_{q=1}^{K} P(w_t|z_q)P(z_q|h_{t-1})} + \frac{i}{t+1} P(z_k|h_{t-1}) \tag{9}$$

#### *2.2.2. LDA-adapted model*

The main difference in the definition of LDA from PLSA is that LDA model integrates over a prior distribution. This integral does not have an analytical solution and LDA uses Variational Bayes

(VB) to train its probabilities; see [6]. VB is an approximate Bayesian technique that uses auxiliary distributions to approximate the model with a more feasible one. It optimises a lower bound of the training data log-likelihood.

The LDA training uses a Dirichlet prior and looks for a most likely set of parameters under this prior. On the other hand, PLSA training searches for the most likely parameters with no prior. Therefore, LDA should be more robust on small data. However unlike PLSA, LDA uses an approximate training algorithm.

To test an LDA-based LM, the N-gram probability was combined with the LDA in the same way as it is combined with PLSA in (3). This is different from the linear interpolation used in the literature [6]. LDA is similar to PLSA in that it can be decomposed into a sum of hidden variable-dependent distributions (4).

Both PLSA and LDA use MAP adaptation to adapt their topic weights in testing. In the LDA setting, the topic weights are again fixed for a given segment. It was not found that adding the segments following the current segment into the history helped the LDA model.

# 3. Experimental evaluation

## 3.1. Recognition system

The LM model adaptation was tested by rescoring lattices produced by an extension of CU-HTK Mandarin system described in [5]. This section provides a brief summary of the system used in the experiments.

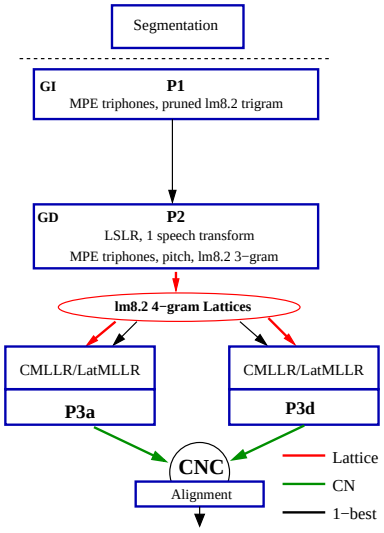


Figure 1: BN-M multi-pass and multi-branch 10xRT system.

The extended BN Mandarin CU-HTK system operates in three passes through the input data. An initial transcript is produced with gender independent models (P1 stage) followed with lattice generation using gender-dependent models (P2 stage). The P2 stage is evaluated at two levels: one-best path through the fourgram lattices (fg) and one-best path in the confusion nets produced from these lattices (fg-cn). The P2 fourgram lattices are used for acoustic re-scoring in two parallel branches (P3 stage): gender dependent (branch **P3d**; MPE-MAP Gaussianised system) and speaker dependent models (branch **P3a**; MPE-SAT Gaussianised system). Confusion nets from these two branches are combined, aligned and the one-best path is the final system output.

The acoustic model of this system was trained on 165 hours of BN data and 20 hours of BC data. The language model is described in section 3.2. in more detail.

## 3.2. Baseline language model

All the N-gram models used 68k open vocabulary and modified Kneser-Ney smoothing. This vocabulary includes 12k English words because there are a number of English words in Mandarin transcripts (1-2%).

| Component | Training data |
|---|---|
| BNM+ENG | mix of Mandarin and English newspapers, broadcast news, web,closed captions [550+1400] |
| BCM | broadcast conversations transcripts [0.288] |

Table 1: Training data and the structure of our baseline model. The text sizes in square brackets are given in millions of words.

The BNM+ENG:BCM interpolation weights were fixed based on preliminary experiments to 0.9:0.1. The weights were set so that they improved the performance on a BC test set as much as possible without changing the performance on BN test sets.

## 3.3. Experimental setup

One decision that needed to be done was at what part of the system the LM adaptation should happen. The later it is the better the hypothesis for adapting the model and therefore potentially the bigger improvement from adapting the LM. On the other hand, the earlier the model is adapted the more stages of the system will use this improved language model and therefore the bigger impact the adaptation may have on the system.

The P2 stage hypothesis was used as a supervision for the LM adaptation. First, the P2 fourgram lattices were produced using the baseline language model. It was the one-best path from these lattices that served as the supervision for adaptation. Then EM was run on this supervision to optimise the interpolation weights of the N-gram model. For PLSA and LDA-based language models, the supervision was used as history to adapt the aspect weights. After optimising the weights, the LM scores in the P2 fourgram lattices were replaced with LM scores of the adapted LM. The system then performed the P3 stage using these new fourgram lattices with adapted LM scores.

## 3.4. Results

The tests were carried out on a BC test set (dev05bcm: 3 hours, 30k words, 5 shows) and two BN test sets (eval04: 1.3 hours, 11k words, 3 shows; dev04f: 0.5 hours, 5k words, 4 shows).

First, the interpolation weights BNM+ENG:BCM were adapted with EM on the supervision (second pass hypothesis) to find out if they differ sufficiently for BN and BC shows to distinguish between the BN and BC data. The show-specific BCM interpolation weights were between 0.389 and 0.466 on all BC shows and between 0 and 0.204 (only one weight bigger than 0.016) on all BN shows. This indicates that it is possible to find a threshold (0.3 in our case) that will distinguish between the BN and BC shows with a margin on both sides (about 0.09 in our case).

As there is much less BC training data than BN data available, the interpolation of BC and BN models with adapted interpolation

weights was used for both BN and BC shows. Note that if there were well-trained models for both BN and BC available it would be possible to switch between these two models without the need for interpolating them. The interpolation weights were fixed for the whole test set. This meant that it was necessary to interpolate the models only once per test set and not for each individual show. The results with show-specific models were the same as with the test-set-specific models.

As can be seen in Table 2, the dynamic mixture weights improved the performance on the BC shows (by 1.1% abs.) but they did not improve the performance on the BN shows. This suggests that it is not necessary to use the interpolation of BC/BN models for BN data. This is because the BN model is trained on enough BN data and adding BC LM does not add any value.

| Test set | baseline fixed weights | N-gram adapt dynamic weights |
|---|---|---|
| dev05bcm (BC) | 25.6 | 24.5 |
| eval04 (BN) | 14.7 | 14.8 |
| dev04f (BN) | 6.4 | 6.5 |

Table 2: P3 CERs for BC and BN test sets.

From the first two columns of Table 3 that contrast a pure BN LM with the baseline trained on both BC and BN data, it can be seen that using a pure BN LM on BC data leads to a large increase of CER. This clearly demonstrates the BN vs. BC mismatch.

As the next step, it was investigated if more sophisticated adaptive techniques improve the performance further. The last two columns of Table 3 present the results for PLSA and LDA adaptations. Both PLSA and LDA models were trained for a shortlist of 14k unique words occurring in the BC training data. All the words from the 68k word list outside this 14k shortlist were assigned N-gram probabilities. In both cases, ten values of the latent variable and ten training iterations were used.

| Phase | fixed weights | | dynamic weights | | |
|---|---|---|---|---|---|
| | no BC | baseline | N-gram | PLSA | LDA |
| fg | 29.5 | 27.8 | 26.7 | 26.6 | 26.6 |
| fg-cn | 28.4 | 26.8 | 25.9 | 25.8 | 25.8 |
| P3 | 27.4 | 25.6 | 24.5 | 24.3 | 24.3 |

Table 3: P2, P3 stage dev05bcm CERs. PLSA and LDA models are combined with the dynamic weights N-gram.

It can be seen in Table 3 that at the P3 stage both PLSA and LDA give the same performance (24.3%) with 1.3% abs. CER gain over the baseline which is an additional 0.2% absolute gain over the N-gram adaptation. At the P2 stage, the LM adaptation brings 0.9% or 1% absolute gain.

Furthermore, the performance of PLSA and LDA models without the N-gram adaptation was investigated. As can be seen in Tables 4 and 3, PLSA performed the same with fixed and dynamic weights after the second pass. On the other hand, LDA performs better when combined with dynamic N-gram model. Table 3 also shows that PLSA adaptation only (25.8%) performed 0.1% CER better than the N-gram adaptation only (25.9%) and LDA performed 0.1% CER worse. Overall the aspect-based adaptation and the N-gram adaptation performed similarly. Combing the aspect-based and N-gram adaptations brings an additional gain over using only one of them.

After additional 900k BC training transcripts became available, the amount of the BC training data increased to 1.2M. Table 5 shows the impact of the increase of the BC training data on LM adaptation. The additional training data alone lead to a better N-gram as expected. Also it can be seen that after adding more BC data, the N-gram adaptation still brings a gain of 1% CER absolute.

| Phase | baseline | dynamic N-gram | PLSA | LDA |
|---|---|---|---|---|
| fg | 27.8 | 26.7 | 26.5 | 26.8 |
| fg-cn | 26.8 | 25.9 | 25.8 | 26.0 |

Table 4: P2 stage dev05bcm CERs. PLSA and LDA models are combined with the baseline (fixed interpolation weights).

| | fixed weights | | dyn. weights |
|---|---|---|---|
| | baseline | more data | more data |
| fg | 27.8 | 27.5 | 26.1 |
| fg-cn | 26.8 | 26.4 | 25.4 |

Table 5: P2 CERs for LM with more in-domain training data.

## 4. Conclusions

It was found that the N-gram adaptation helps to improve the performance on the BC transcription task. The specificity of this task is having a mix of data with very different styles. Also at present, there is a small amount of LM training data for the BC style.

The gains from the LM adaptation stay at the same level at different stages of the system for all investigated methods. Both N-gram adaptation and aspect-based models perform similarly and the combination of a dynamic N-gram with an aspect model brings an additional gain. In contrast to previously published perplexity comparisons of PLSA and LDA [1, 2], there was not any substantial difference in the character error rate between these two methods.

## 5. ACKNOWLEDGEMENTS

This work was supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

## 6. References

[1] Blei, D., Ng, A. and Jordan, M.,"Latent Dirichlet Allocation", Journal of Machine Learning Research, 2003.

[2] Brants, T., "Test Data Likelihood for PLSA Models", Information Retrieval, 2005, volume 8, number 2, pp 181–196.

[3] Gildea, D. and Hofmann, T., "Topic-based language models using EM", Proc. of Eurospeech, 6, 1999.

[4] Mrva, D. and Woodland, P. C., "A PLSA-based Language Model for Conversational Telephone Speech", Proc. of Interspeech, 2004.

[5] Sinha, R., Gales, M. J. F., Kim, D. Y., Liu, X. A., Sim, K.C. and Woodland, P. C., "The CU-HTK Mandarin Broadcast News Transcription System", Proc. of the International Conference on Acoustics, Speech, and Signal Processing, 2006.

[6] Tam, Y. C. and Schultz, T., "Dynamic language model adaptation using variational Bayes inference", Proc. of Interspeech, 2005, pp 5-8.

[7] Chien, J. T., Wu, M. S. and Wu, C. S., "Bayesian Learning for Latent Semantic Analysis", Proc. of Interspeech, 2005.