Dual Language Models for
Code Switched Speech Recognition

Saurabh Garg, Tanmay Parekh, Preethi Jyothi

Department of Computer Science and Engineering, Indian Institute of Technology Bombay

{saurabhgarg,tanmayb,pjyothi}@cse.iitb.ac.in

Abstract

In this work, we present a simple and elegant approach to language modeling for bilingual code-switched text. Since code-switching is a blend of two or more different languages, a standard bilingual language model can be improved upon by using structures of the monolingual language models. We propose a novel technique called *dual language models*, which involves building two complementary monolingual language models and combining them using a probabilistic model for switching between the two. We evaluate the efficacy of our approach using a conversational Mandarin-English speech corpus. We prove the robustness of our model by showing significant improvements in perplexity measures over the standard bilingual language model without the use of any external information. Similar consistent improvements are also reflected in automatic speech recognition error rates.

Index Terms: Code-switching, language modeling, speech recognition

1. Introduction

Code-switching is a commonly occurring phenomenon in multilingual communities, wherein a speaker switches between languages within the span of a single utterance. Code-switched speech presents many challenges for automatic speech recognition (ASR) systems, in the context of both acoustic models and language models. Mixing of dissimilar languages leads to loss of structure, which makes the task of language modeling more difficult. Our focus in this paper is on building robust language models for code-switched speech from bilingual speakers.

A naïve approach towards this problem would be to simply use a bilingual language model. However, the complexity of a full-fledged bilingual language model is significantly higher than that of two monolingual models, and is unsuitable in a limited data setting. More sophisticated approaches relying on translation models have been proposed to overcome this challenge (see Section 2), but they rely on external resources to build the translation model. In this paper, we introduce an alternate – and simpler – approach to address the challenge of limited data in the context of code-switched text without use of any external resources.

At the heart of our solution is a *dual language model* (DLM) that has roughly the complexity of two monolingual language models combined. A DLM combines two such models and uses a probabilistic model to switch between them. Its simplicity makes it amenable for generalization in a low-data context. Further there are several other benefits of using DLMs. (1) The DLM construction does not rely on any prior information about the underlying languages. (2) Since the structure of our combined model is derived from monolingual language models, it can be implemented as a finite-state machine and easily

incorporated within an ASR system. (3) The monolingual language model for the primary language can be trained further with large amounts of monolingual text data (which is easier to obtain compared to code-switched text).

Our main contributions can be summarized as follows:
- We formalize the framework of DLMs (Section 3).
- We show significant improvements in perplexity using DLMs when compared against smoothed n-gram language models estimated on code-switched text (Section 4.3). We provide a detailed analysis of DLM improvements (Section 5).
- We evaluate DLMs on the ASR task. DLMs capture sufficient complementary information which we leverage to show improvements on error rates. (Section 4.4).

2. Related Work

Prior work on building ASR systems for code-switched speech can be broadly categorized into two sets of approaches: (1) Detecting code-switching points in an utterance, followed by the application of monolingual acoustic and language models to the individual segments [1, 2, 3]. (2) Employing a universal phone set to build acoustic models for the mixed speech and pairing it with standard language models trained on code-switched text [4, 5, 6, 7, 8].

There have been many past efforts towards enhancing the capability of language models for code-switched speech using additional sources of information such as part-of-speech (POS) taggers and statistical machine translation (SMT) systems. Yeh et al. [7] employed class-based n-gram models that cluster words from both languages into classes based on POS and perplexity-based features. Vu et al. [9] used an SMT system to enhance the language models during decoding. Li et al. [10] propose combining a code-switch boundary predictor with both a translation model and a reconstruction model to build language models. (Solorio et. al. [11] were one of the first works on learning to predict code-switching points.) Adel et al. [12] investigated how to effectively use syntactic and semantic features extracted from code-switched data within factored language models. Combining recurrent neural network-based language models with such factored language models has also been explored [13].

3. Dual language models

We define a *dual language model* (DLM) to have the following 2-player game structure. A sentence (or more generally, a sequence of tokens) is generated via a co-operative game between the two players who take turns. During its turn a player generates one or more words (or tokens), and either terminates the sentence or transfers control to the other player. Optionally, while transferring control, a player may send additional information to the other player (e.g., the last word it produced), and

Given two language models $\mathcal{L}_1$ and $\mathcal{L}_2$ with conditional probabilities P_1 and P_2 that satisfy the following conditions:

$$P_1[\langle/\mathtt{s}\rangle \mid \langle\mathtt{s}\rangle] = P_2[\langle/\mathtt{s}\rangle \mid \langle\mathtt{s}\rangle] = 0 \qquad (1) \qquad\qquad P_1[\langle\mathtt{sw}\rangle \mid \langle\mathtt{s}\rangle] + P_2[\langle\mathtt{sw}\rangle \mid \langle\mathtt{s}\rangle] = 1 \qquad (2)$$

$$P_1[\langle\mathtt{sw}\rangle \mid \langle\mathtt{sw}\rangle] = P_2[\langle\mathtt{sw}\rangle \mid \langle\mathtt{sw}\rangle] = 0 \qquad (3) \qquad\qquad P_1[\langle/\mathtt{s}\rangle \mid \langle\mathtt{sw}\rangle] = P_2[\langle/\mathtt{s}\rangle \mid \langle\mathtt{sw}\rangle] = 0 \qquad (4)$$

We define a combined language model $\mathcal{L}$, with conditional probabilities P, as follows:

$$P[w' \mid w] = \begin{cases} P_1[w' \mid \langle\mathtt{s}\rangle] & \text{if } w' \in \mathrm{V}_1 \\ P_2[w' \mid \langle\mathtt{s}\rangle] & \text{if } w' \in \mathrm{V}_2 \\ 0 & \text{if } w' = \langle/\mathtt{s}\rangle \end{cases} \qquad\qquad \text{for } w = \langle\mathtt{s}\rangle$$

$$P[w' \mid w] = \begin{cases} P_1[w' \mid w] & \text{if } w' \in \mathrm{V}_1 \cup \{\langle/\mathtt{s}\rangle\} \\ P_1[\langle\mathtt{sw}\rangle \mid w] \cdot P_2[w' \mid \langle\mathtt{sw}\rangle] & \text{if } w' \in \mathrm{V}_2 \end{cases} \qquad\qquad \text{for } w \in \mathrm{V}_1$$

$$P[w' \mid w] = \begin{cases} P_2[w' \mid w] & \text{if } w' \in \mathrm{V}_2 \cup \{\langle/\mathtt{s}\rangle\} \\ P_2[\langle\mathtt{sw}\rangle \mid w] \cdot P_1[w' \mid \langle\mathtt{sw}\rangle] & \text{if } w' \in \mathrm{V}_1 \end{cases} \qquad\qquad \text{for } w \in \mathrm{V}_2$$

Figure 1: *Definition of a bigram-based DLM for code-switched text.*

also may retain some state information (e.g., cached words) for its next turn. At the beginning of the game one of the two players is chosen probabilistically.

In the context of code-switched text involving two languages, we consider a DLM wherein the two players are each in charge of generating tokens in one of the two languages. Suppose the two languages have (typically disjoint) vocabularies V_1 and V_2. Then the alphabet of the output tokens produced by the first player in a single turn is $\mathrm{V}_1 \cup \{\langle\mathtt{sw}\rangle, \langle/\mathtt{s}\rangle\}$, $\langle\mathtt{sw}\rangle$ denotes the *switching* – i.e., transferring control to the other player – and $\langle/\mathtt{s}\rangle$ denotes the end of sentence, terminating the game. We shall require that a player produces at least one token before switching or terminating, so that when $\mathrm{V}_1 \cap \mathrm{V}_2 = \emptyset$, any non-empty sentence in $(\mathrm{V}_1 \cup \mathrm{V}_2)^*$ uniquely determines the sequence of corresponding outputs from the two players when the DLM produces that sentence. (Without this restriction, the players can switch control between each other arbitrarily many times, or have either player terminate a given sentence.)

In this paper, we explore a particularly simple DLM that is constructed from two given LMs for the two languages. More precisely, we shall consider an LM $\mathcal{L}_1$ which produces $\langle/\mathtt{s}\rangle$-terminated strings in $(\mathrm{V}_1 \cup \{\langle\mathtt{sw}\rangle\})^*$ where $\langle\mathtt{sw}\rangle$ indicates a span of tokens in the *other* language (so multiple $\langle\mathtt{sw}\rangle$ tokens cannot appear adjacent to each other), and symmetrically an LM $\mathcal{L}_2$ which produces strings in $(\mathrm{V}_2 \cup \{\langle\mathtt{sw}\rangle\})^*$. In Section 4.2, we will describe how such monolingual LMs can be constructed from code-switched data. Given $\mathcal{L}_1$ and $\mathcal{L}_2$, we shall splice them together into a simple DLM (in which players do not retain any state between turns, or transmit state information to the other player at the end of a turn). Below we explain this process which is formally described in Fig. 1 (for bi-gram language models).

We impose conditions (1)-(4) on the given LMs. Condition (1) which disallows empty sentences in the given LMs (and the resulting LM) is natural, and merely for convenience. Condition (2) states the requirement that $\mathcal{L}_1$ and $\mathcal{L}_2$ agree on the probabilities with which each of them gets the first turn. Conditions (3) and (4) require that after switching at least one token should be output before switching again or terminating. If the two LMs are trained on the same data as described in Section 4.2, all these conditions would hold.

To see that $P[w' \mid w]$ defined in Fig. 1 is a well-defined probability distribution, we check that $\sum_{w'} P[w'|w] = 1$ for all three cases of w, where the summation is over $w' \in \mathrm{V}_1 \cup$
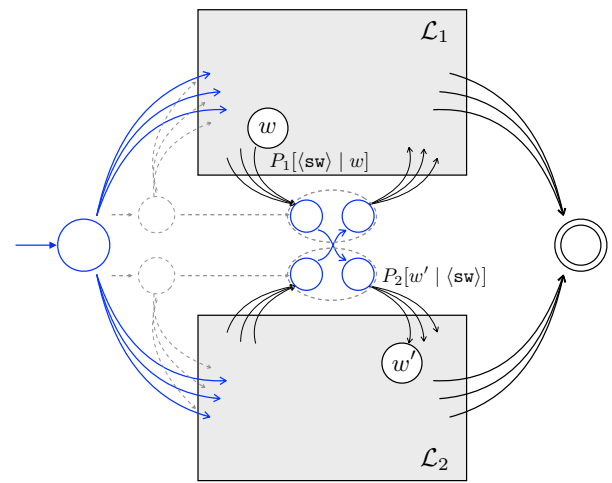

Figure 2: *DLM using two monolingual LMs, $\mathcal{L}_1$ and $\mathcal{L}_2$, implemented as a finite-state machine.*

$\mathrm{V}_2 \cup \{\langle/\mathtt{s}\rangle\}$. When $w = \langle\mathtt{s}\rangle$, $\sum_{w'} P[w'|w]$ equals

$$\sum_{w' \in \mathrm{V}_1} P_1[w'|\langle\mathtt{s}\rangle] + \sum_{w' \in \mathrm{V}_2} P_2[w'|\langle\mathtt{s}\rangle]$$
$$= (1 - P_1[\langle\mathtt{sw}\rangle \mid \langle\mathtt{s}\rangle]) + (1 - P_2[\langle\mathtt{sw}\rangle \mid \langle\mathtt{s}\rangle]) = 1$$

where the first equality is from (1) and the second equality is from (2).

When $w \in \mathrm{V}_1$, $\sum_{w'} P[w'|w]$ is

$$\sum_{w' \in \mathrm{V}_1 \cup \langle/\mathtt{s}\rangle} P_1[w'|w] + P_1[\langle\mathtt{sw}\rangle \mid w] \sum_{w' \in \mathrm{V}_2} P_2[w'|\langle\mathtt{sw}\rangle]$$
$$= \sum_{w' \in \mathrm{V}_1 \cup \langle/\mathtt{s}\rangle} P_1[w'|w] + P_1[\langle\mathtt{sw}\rangle \mid w] = 1.$$

The case of $w \in \mathrm{V}_2$ follows symmetrically.

Figure 2 illustrates how to implement a DLM as a finite-state machine using finite-state machines for the monolingual bigram LMs, $\mathcal{L}_1$ and $\mathcal{L}_2$. The start states in both LMs, along with all the arcs leaving these states, are deleted; a new start state and end state is created for the DLM with accompanying

arcs as shown in Figure 2. The two states maintaining information about the $\langle sw \rangle$ token can be split and connected, as shown in Figure 2, to create paths between $\mathcal{L}_1$ and $\mathcal{L}_2$.

4. Experiments and Results

4.1. Data description

We make use of the SEAME corpus [14] which is a conversational Mandarin-English code-switching speech corpus.

Preprocessing of data. Apart from the code-switched speech, the SEAME corpus comprises of a) words of foreign origin (other than Mandarin and English) b) incomplete words c) unknown words labeled as $\langle unk \rangle$, and d) mixed words such as bleach跟, cause就是, etc.. Since it was difficult to obtain pronunciations for these words, we removed utterances that contained any of these words. A few utterances contained markers for non-speech sounds like laughing, breathing, etc. Since our focus in this work is to investigate language models for code-switching, ideally without the interference of these non-speech sounds, we excluded these utterances from our task.

Data distribution. We construct training, development and test sets from the preprocessed SEAME corpus data using a 60-20-20 split. Table 1 shows detailed statistics of each split. The development and evaluation sets were chosen to have 37 and 30 random speakers each, disjoint from the speakers in the training data.[1] The out-of-vocabulary (OOV) rates on the development and test sets are 3.3% and 3.7%, respectively.

	Train	Dev	Test
# Speakers	90	37	30
Duration (hrs)	56.6	18.5	18.7
# Utterances	54,020	19,976	19,784
# Tokens	539,185	195,551	196,462

Table 1: *Statistics of the dataset*

4.2. Monolingual LMs for the DLM construction

Given a code-switched text corpus D, we will derive two complementary corpora, D_1 and D_2, from which we construct bigram models $\mathcal{L}_1$ and $\mathcal{L}_2$ as required by the DLM construction in Figure 1, respectively. In D_1, spans of tokens in the second language are replaced by a single token $\langle sw \rangle$. D_2 is constructed symmetrically. Standard bigram model construction on D_1 and D_2 ensures conditions (1) and (2) in Figure 1. The remaining two conditions may not naturally hold: Even though the data in D_1 and D_2 will not have consecutive $\langle sw \rangle$ tokens, smoothing operations may assign a non-zero probability for this; also, both LMs may assign non-zero probability for a sentence to end right after a $\langle sw \rangle$ token, corresponding to the sentence having ended with a non-empty span of tokens in the other language. These two conditions are therefore enforced by reweighting the LMs.

4.3. Perplexity experiments

We used the SRILM toolkit [15] to build all our LMs. The baseline LM is a smoothed bigram LM estimated using the code-switched text which will henceforth be referred to as *mixed LM*. Our DLM was built using two monolingual bigram LMs. (The choice of bigram LMs instead of trigram LMs will be justified later in Section 5). Table 2 shows the perplexities on the validation and test sets using both Good Turing [16] and

Kneser-Ney [17] smoothing techniques. DLMs clearly outperform mixed LMs on both the datasets. All subsequent experiments use Kneser-Ney smoothed bigram LMs as they perform better than the Good Turing smoothed bigram LMs.

Smoothing	Dev		Test	
Technique	mixed LM	DLM	mixed LM	DLM
Good Turing	338.2978	**329.1822**	384.5164	**371.1112**
Kneser-Ney	329.6725	**324.9268**	376.0968	**369.9355**

Table 2: *Perplexities on the dev/test sets using mixed LMs and DLMs with different smoothing techniques.*

We also evaluate perplexities by reducing the amount of training data to $\frac{1}{2}$ or $\frac{1}{3}$ of the original training data (shown in Table 3). As we reduce the training data, the improvements in perplexity of DLM over mixed LM further increase, which validates our hypothesis that DLMs are capable of generalizing better. Section 5 elaborates this point further.

Training	Dev		Test	
data	mixed LM	DLM	mixed LM	DLM
Full	329.6725	**324.9268**	376.0968	**369.9355**
1/2	362.0966	**350.5860**	400.5831	**389.7618**
1/3	368.6205	**356.012**	408.562	**394.2131**

Table 3: *Kneser-Ney smoothed bigram dev/test set perplexities using varying amounts of training data*

4.4. ASR experiments

All the ASR systems were built using the Kaldi toolkit [18]. We used standard mel-frequency cepstral coefficient (MFCC)+delta+double-delta features with feature space maximum likelihood linear regression (fMLLR) [19] transforms to build speaker-adapted triphone models with 4200 tied-state triphones, henceforth referred to as "SAT" models. We also build time delay neural network (TDNN [20])-based acoustic models using i-vector based features (referred to as "TDNN+SAT"). Finally, we also re-scored lattices generated by the "TDNN+SAT" model with an RNNLM [21] (referred to as "RNNLM Rescoring"), trained using Tensorflow [22] on the SEAME training data. [2] We trained a single-layer RNN with 200 hidden units in the LSTM [23] cell.

The pronunciation lexicon was constructed from CMU-dict [24] and THCHS30 dictionary [25] for English and Mandarin pronunciations, respectively. Mandarin words that did not appear in THCHS30 were mapped into Pinyin using a freely available Chinese to Pinyin converter.[3] We manually merged the phone sets of Mandarin and English (by mapping all the phones to IPA) resulting in a phone inventory of size 105.

To evaluate the ASR systems, we treat English words and Mandarin characters as separate tokens and compute token error rates (TERs) as discussed in [9]. Table 4 shows TERs on the dev/test sets using both mixed LMs and DLMs. DLM performs better or on par with mixed LM and at the same time, captures a significant amount of complementary information which we leverage by combining lattices from both systems. The improvements in TER after combining the lattices are statistically significant (at $p < 0.001$) for all three systems, which justifies our claim of capturing complementary information. Trigram mixed LM performance was worse than bigram mixed

---

[1] We note that choosing fewer speakers in the development and test sets led to high variance in the observed results.

[2] This rescoring was implemented using the `tfrnnlm` binary provided by Kaldi [18] developers.

[3] `https://www.chineseconverter.com/en/convert/chinese-to-pinyin`

ASR system	Data	mixed LM	DLM	combined
SAT	Dev	45.59	45.59	**44.93***
	Test	47.43	47.48	**46.96***
TDNN+SAT	Dev	35.20	35.26	**34.91***
	Test	37.42	37.35	**37.17***
RNNLM Rescoring	Dev	34.21	34.11	**33.85***
	Test	36.64	36.52	**36.37***

Table 4: *TERs using mixed LMs and DLMs*

LM; hence we adopted the latter in all our models (further discussed in Section 5). This demonstrates that obtaining significant performance improvements via LMs on this task is very challenging. Table 5 shows all the TER numbers by utilizing only $\frac{1}{2}$ of the total training data. The combined models continue to give significant improvements over the individual models. Moreover, DLMs consistently show improvements on TERs compared to mixed LMs in the $\frac{1}{2}$ training data setting.

ASR system	Data	mixed LM	DLM	combined
SAT	Dev	48.48	48.17	**47.67**[1]
	Test	49.07	49.04	**48.52***
TDNN+SAT	Dev	40.59	40.48	**40.12***
	Test	41.34	41.32	**41.13***
RNNLM Rescoring	Dev	40.20	40.09	**39.84***
	Test	40.98	40.90	**40.72***

Table 5: *TERs with $\frac{1}{2}$ training data*

5. Discussion

Code-switched data corpora tend to exhibit very different linguistic characteristics compared to standard monolingual corpora, possibly because of the informal contexts in which code-switched data often occurs, and also possibly because of the difficulty in collecting such data. It is possible that the gains made by our language model are in part due to such characteristics of the corpus we use, SEAME. (We note that this corpus is by far the most predominant one used to benchmark speech recognition techniques for code-switched speech.)

In this section we analyze the SEAME corpus and try to further understand our results in light of its characteristics.

Code-switching boundaries. Code-switched bigrams with counts of ≤ 10 occupy 87.5% of the total number of code-switched bigrams in the training data. Of these, 55% of the bigrams have a count of 1. This suggests that context across code-switching boundaries cannot significantly help a language model built from this data. Indeed, the DLM construction in this work discards such context, in favor of a simpler model.

n**-gram token distribution.** We compare the unigram distribution of a code-switched corpus (SEAME) with a standard monolingual corpus (PTB [26]). A glaring difference is observed in their distributions (Figure 3-a) with significantly high occurrence of less-frequent unigrams in the code-switched corpus, which makes them rather difficult to capture using standard n-gram models (which often fall back to a unigram model). The DLM partially compensates for this by emulating a "class-based language model," using the only class information readily available in the data (namely, the language of each word).

Illustrative examples. Below, we analyze perplexities of the mixed LM and the DLM on some representative sentences from the SEAME corpus, to illustrate how the performances of the two models compare.

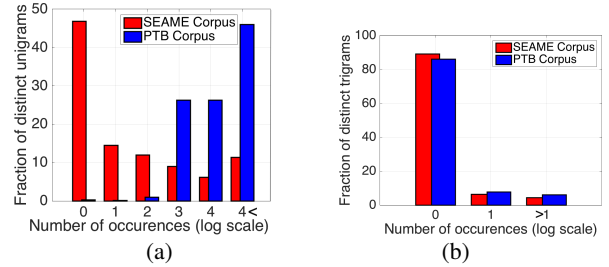[1]statistically significant improvement (at $p < 0.001$)

(a) (b)

Figure 3: *Comparison of fraction of data vs. frequency of n-grams in code-mixed text (SEAME) and monolingual English (PTB) text. (The X-axis uses the log2 scale; 1 indicates unigrams with frequencies $\leq 2^1$, etc.)*

Sentence	*Mixed LM* perplexity	*DLM* perplexity
我们 的 total 是 五十七	920.8	720.4
哦 我 没有 meeting 了	92.2	75.9
okay kay 让 我 拿出 我 的 calculator	1260.3	1284.6
the roomie lives in serangoon right	2302.1	1629.3
oh 他 拿 third class 他 差 一点点 他 的 f. y. p. screwed up 他 拿 到 b minus c plus	299.7	257.1

We observe that when less frequent words appear at switching points (like total, meeting, etc.), the DLM outperforms the mixed LM by a significant margin as illustrated in the first two sentences above. In cases of highly frequent words occurring at switching points, the DLM performs on par with or slightly worse than the mixed LM, as seen in the case of the third sentence. The DLM also performs slightly better within long stretches of monolingual text as seen in the fourth sentence. On the final sentence, which has multiple switches and long stretches of monolingual text, again the DLM performs better. As these examples illustrate, DLMs tend to show improved performance at less frequent switching points and within long stretches of monolingual text.

Effect of Trigrams. In standard monolingual datasets, trigram models consistently outperform bilingual models. However, in the SEAME corpus we did not find a pronounced difference between a bigram and a trigram model. This could be attributed to the fact that the number of highly frequent trigrams in our corpus is lower in comparison to that in the PTB dataset (Figure 3-b). As such, we have focused on bigram LMs in this work.

6. Conclusions

We introduced DLMs and showed robust improvements over mixed LMs in perplexity for code-switched speech. While the performance improvements for the ASR error rates are modest, they are achieved without the aid of any external language resources and without any computational overhead. We observe significant ASR improvements via lattice combination of DLMs and the standard mixed LMs. Future directions include investigating properties of code-switched text which can be incorporated within DLMs, using monolingual data to enhance each DLM component and demonstrating the value of DLMs for multiple code-switched language pairs.

7. References

[1] J. Y. Chan, P. Ching, T. Lee, and H. M. Meng, "Detection of language boundary in code-switching utterances by bi-phone probabilities," in *Proceedings of International Symposium on Chinese Spoken Language Processing*. IEEE, 2004, pp. 293–296.

[2] D.-C. Lyu and R.-Y. Lyu, "Language identification on code-switching utterances using multiple cues," in *Proceedings of Interspeech*, 2008.

[3] C.-J. Shia, Y.-H. Chiu, J.-H. Hsieh, and C.-H. Wu, "Language boundary detection and identification of mixed-language speech based on MAP estimation," in *Proceedings of ICASSP*, vol. 1. IEEE, 2004, pp. I–381.

[4] D. Imseng, H. Bourlard, M. M. Doss, and J. Dines, "Language dependent universal phoneme posterior estimation for mixed language speech recognition," in *Proceedings of ICASSP*, 2011, pp. 5012–5015.

[5] Y. Li, P. Fung, P. Xu, and Y. Liu, "Asymmetric acoustic modeling of mixed language speech," in *Proceedings of ICASSP*, 2011, pp. 5004–5007.

[6] K. Bhuvanagiri and S. Kopparapu, "An approach to mixed language automatic speech recognition," *Proceedings of Oriental COCOSDA, Kathmandu, Nepal*, 2010.

[7] C. F. Yeh, C. Y. Huang, L. C. Sun, and L. S. Lee, "An integrated framework for transcribing Mandarin-English code-mixed lectures with improved acoustic and language modeling," in *Proceedings of Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on*. IEEE, 2010, pp. 214–219.

[8] S. Yu, S. Hu, S. Zhang, and B. Xu, "Chinese-English bilingual speech recognition," in *Proceedings of Natural Language Processing and Knowledge Engineering*. IEEE, 2003, pp. 603–609.

[9] N. T. Vu, D.-C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, "A first speech recognition system for Mandarin-English code-switch conversational speech," in *Proceedings of ICASSP*. IEEE, 2012, pp. 4889–4892.

[10] Y. Li and P. Fung, "Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints," in *Proceedings of ICASSP*. IEEE, 2013, pp. 7368–7372.

[11] T. Solorio and Y. Liu, "Learning to predict code-switching points," in *Proceedings of EMNLP*, 2008, pp. 973–981.

[12] H. Adel, N. T. Vu, K. Kirchhoff, D. Telaar, and T. Schultz, "Syntactic and semantic features for code-switching factored language models," *Proceedings of IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 431–440, 2015.

[13] H. Adel, D. Telaar, N. T. Vu, K. Kirchhoff, and T. Schultz, "Combining recurrent neural networks and factored language models during decoding of code-switching speech," in *Proceedings of Interspeech*, 2014.

[14] D.-C. Lyu, T.-P. Tan, E.-S. Chng, and H. Li, "An analysis of a Mandarin-English code-switching speech corpus: SEAME," *Proceedings of Age*, vol. 21, pp. 25–8, 2010.

[15] A. Stolcke, "SRILM – an extensible language modeling toolkit." in *Proceedings of Interspeech*, 2002.

[16] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *Proceedings of IEEE Transactions on Acoustics, Speech, and Signal Processing*. IEEE, 1987.

[17] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of ICASSP*. IEEE, 1995.

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011.

[19] M. J. F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," in *Proceedings of Computer Speech and Language*. ELSEVIER, 1998.

[20] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural network," in *Readings in speech recognition*. ELSEVIER, 1990.

[21] X. Hainan, C. Tongfei, G. Dongji, W. Yiming, L. Ke, G. Nagendra, C. Yishay, P. Daniel, and K. Sanjeev, "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition," in *Proceedings of ICASSP*. IEEE, 2017.

[22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *Proceedings of OSDI*, vol. 16, 2016, pp. 265–283.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] R. L. Weide, "The CMU pronouncing dictionary," *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*, 1998.

[25] Z. Z. Dong Wang, Xuewei Zhang, "THCHS-30 : A free Chinese speech corpus," 2015. [Online]. Available: http://arxiv.org/abs/1512.01882

[26] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.