Improving N-gram Language Modeling for Code-switching Speech Recognition

Zhiping Zeng*, Haihua Xu†, Tze Yuang Chong†, Eng-Siong Chng†, Haizhou Li ‡
* School of Communication & Information Engineering, Shanghai University, Shanghai, China
† School of Computer Science and Engineering, Nanyang Technological University, Singapore
‡ Department of Electrical and Computer Engineering, National University of Singapore

Abstract—**Code-switching language modeling is challenging due to statistics of each individual language, as well as statistics of cross-lingual language are insufficient. To compensate for the issue of statistical insufficiency, in this paper we propose a word-class n-gram language modeling approach of which only infrequent words are clustered while most frequent words are treated as singleton classes themselves. We first demonstrate the effectiveness of the proposed method on our English-Mandarin code-switching SEAME data in terms of perplexity. Compared with the conventional word n-gram language models, as well as the word-class n-gram language models of which entire vocabulary words are clustered, the proposed word-class n-gram language modeling approach can yield lower perplexity on our SEAME *dev* data sets. Additionally, we observed further perplexity reduction by interpolating the word n-gram language models with the proposed word-class n-gram language models. We also attempted to build word-class n-gram language models using third-party text data with our proposed method, and similar perplexity performance improvement was obtained on our SEAME *dev* data sets when they are interpolated with the word n-gram language models. Finally, to examine the contribution of the proposed language modeling approach to code-switching speech recognition, we conducted lattice based n-best rescoring.**

I. Introduction

Language and acoustic modelings are two key factors of a state-of-the-art automatic speech recognition (ASR) system that is in turn an indispensable part for modern spoken language understanding. However, most of present ASR system is only capable of monolingual understanding, which is not desirable on some occasions where code-switching[1] based multilingual conversations frequently occur.

Compared with building of a monolingual ASR system, code-switching based multilingual ASR system building is much harder, at least due to data sharing and data insufficiency issues. Both issues can exist either in acoustic modeling or language modeling. For code-switching based acoustic modeling, [1]–[3] proposed using IPA phone set for the benefit of better data sharing. While [4]–[6] pursued language dependent phone sets for each individual languages, but phone or senone merging were attempted. In recent years, as DNN acoustic modelling framework comes to popularity, performance of Code-switching Speech Recognition (CSSR) has been improved significantly [7], thanks to the capability of DNN acoustic modelling , as well as its inherent capacity for data sharing across different languages.

Code-switching incurs severe data scarcity problem in language modeling, particularly for the n-grams which consist of language transitions, e.g. “ 一起 去 canteen” and “then 我们 就”, for which obtaining adequate data to learn the respective probabilities is difficult. As majority corpora are monolingual, existing techniques of language models adaption, either data augmentation [8], [9] or model combination [10], [11], cannot be readily applied to improve the language models. To tackle such specific problem, several approaches have been proposed such as incorporating linguistic rules to govern the language transitions between words [12], [13], approximating the probabilities by using features that are more general like POS and word-class [14]–[17], and projecting words into continuous space to achieve better generalization [18]–[20]. Overall, these approaches derive complementary information about the training data to improve the language model.

Moreover, code-switching language is usually spoken in spontaneous which contains certain speaking habits that are specific to a group of speakers, two corpora of the same code-switching language collected in different regions might possess different syntactic structures. For example, although Mandarin-English code-switching language is commonly spoken in South-East Asia and Mainland China, how speakers switch languages are different, e.g. it is more common for the speaker in Mainland China to say “ 挺 tough 的” than in South-East Asia. Combining two models of speaking style mismatch by using the traditional interpolation methods results suboptimal model.

In this paper, we are aiming to alleviate data sparsity issue in code-switching using word-class based language modeling method. To this end, we propose a restricted word clustering method in which only those infrequent words are clustered while those frequent words are dealt with singleton classes themselves. We dubbed this as “restricted” word clustering method since some kind of “raw” prior knowledge is considered when we cluster words into classes. This is obviously contrasted with conventional ones where the entire word vocabulary are clustered using data-driven based clustering method. Although this is straightforward when no third-party data is available, its effectiveness lies in being capable of extracting pertained syntactic structures for those infrequent

[1] In this paper, code-switching refers to different language transition phenomenon, either in a speech utterance or between utterances. To be simplified, we only consider English-Mandarin language code-switching phenomenon.

n-grams from third-party data while keeping those frequent n-grams marginally affected when third-party data is employed.

We demonstrate the effectiveness of the proposed method in two configurations. that is, the case of with or without third-party data employed. First, We found the "restricted" word clustering method yields improved perplexity results when it is interpolated with word n-gram language models when no third-party data is used at all. It also performs better compared with the method where overall word vocabulary are clustered. Secondly, when third-party data is available, the advantage of the proposed method is further increased thanks to its amelioration of probability estimation for those infrequent n-grams using more data. Finally, to examine the effectiveness of the proposed method in code-switching speech recognition, we conducted lattice rescoring.

II. Data description

In this work, two categories of data sets are employed for the overall experiments. One is SEAME code-switching data [21], the other category is composed of two third-party (out-domain) data sets that are an English transcription corpus and a Mandarin transcription corpus respectively. Table I depicts the SEAME data set.

TABLE I
Data description of SEAME corpus for language modeling

Data set	Size (M)	Vocabulary (K)	OOV (%)
Train	0.91	29.87	-
dev_{man}	0.07	6.45	2.13
dev_{sge}	0.06	5.36	1.40

The overall text of the three data sets in Table I is from the entire transcription of SEAME corpus. Specifically, they correspond to three data sets of ∼97 hours, ∼8 hours and ∼6 hours with each containing 134, 10 and 10 speakers respectively. In terms of OOV rate in Table I, English and Mandarin words are ∼60% and ∼40% respectively in dev_{man}, while they are ∼70% and ∼30% in dev_{sge} instead. From this perspective, we can see English words are not as well covered as those of Mandarin for the two *dev* sets.

Table II indicates individual language word occurrence rates on average of each utterance for the three data sets. From Table II, we can see Mandarin is dominant in SEAME data. However as seen in Tables I and II the two defined *dev* sets contain different proportions of English and Mandarin data, as dev_{man} is dominated with Mandarin data, while dev_{sge} is dominated with English data. They are defined differently because we want to have more insights on the actual performance change under different code-switching scenarios.

Table III describes the general distribution of the third-party data. The Mandarin text is from the transcription of the LDC2005T32 corpus, which corresponds to ∼190 hours of telephony speech data. The English text is the transcription of the ∼160 hours of Singaporean speech data. We choose them because the two data sets, as well as our SEAME data set, belong to the category of spontaneous speech. Besides, real code-switching data is hard to obtain, but we still want to merge the two data sets to simulate a kind of corpus approximately equivalent to code-switching data by means of word-class based language modeling as will be shown in Section III. However, we notice that both Mandarin and English data sets in Table III are not purely monolingual data set, each itself containing very small portions of code-switching data.

TABLE II
Individual language word occurrence rate on average of utterance in SEAME corpus

Data set	Mandarin (%)	English (%)
Train	59.05	40.95
dev_{man}	65.66	34.34
dev_{sge}	27.93	72.07

In Table III, we also show the vocabulary overlap rate of the two data sets relative to the SEAME data set. We define overlap rate as the portion of the intersected vocabulary between the third-party data and the SEAME data relative to the vocabulary of the SEAME training data. For instance, the overlap rate of the Mandarin data is 35.89%. If we assume the English data it contains can be ignored, then we can conclude only half of the Mandarin vocabulary (refer to Table II) in the SEAME data appears in the third-party Mandarin data. Which means both data sets are obviously different in terms of topic. However, majority of English vocabulary in the SEAME data is covered by the third-party English data. Both data sets are also clearly different, for there are still about 10% of vocabulary in the SEAME training data not covered (refer to Table II).

TABLE III
Data distribution of third-party corpora for language modeling

Data set	Size (M)	Vocabulary (K)	Vocabulary Overlap rate (%)
Mandarin	1.45	47.93	35.89
English	1.41	21.11	32.72

III. Word-class n-gram language modeling

In this section, we review some backgrounds of n-gram language modeling method for the sake of better clarification in what follows.

A. Restricted word-class based Language modeling

Equation 1 shows how word-class based language model estimates the n-gram probabilities:

$$P(w_i|w_{i-n+1},\ldots,w_{i-1}) = P(C(w_i)|C(w_{i-n+1}),\ldots,C(w_{i-1}))P(w_i|C(w_i)) \quad (1)$$

where $C(w_i)$ refers to the class of word w_i. The first term on the right hand side of Equation 1 is estimated according to normal n-gram formula as shown in the following Equation

3, while the second term is estimated with word count in a class to which it belongs. For our restricted word clustering method, when word w_i is a frequent word whose count is above a specified threshold, Equation 1 is changed to:

$$P(w_i|w_{i-n+1},\ldots,w_{i-1}) = P(w_i|C(w_{i-n+1}),\ldots,C(w_{i-1})) \tag{2}$$

This is because word w_i itself is a singleton class, thus $P(w_i|C(w_i)) = 1$. We note that our restricted word-class n-gram language modeling method does not violate the requirements of n-gram language modeling theory.

B. Smoothing probability estimate

N-gram language modeling has been widely used in speech recognition community for long thanks to its simplicity and effectiveness. However, one of main limitations of N-gram language modeling is that it suffers from data sparsity issue given a limited data set. To counter this problem, various discount based backing-off mechanisms were proposed [22]–[24] to smooth the probability distribution. The general principle can be explained with Equation 3 [25].

$$\hat{P}(w_i|w_{i-n+1},\ldots,w_{i-1}) = \begin{cases} \alpha(w_{i-n+1},\ldots,w_{i-1})\hat{P}(w_i|w_{i-n+2},\ldots,w_{i-1}), \\ \qquad\qquad c(w_{i-n+1},\ldots,w_i)=0 \\ d_{c(w_{i-n+1},\ldots,w_i)}\frac{c(w_{i-n+1},\cdots,w_i)}{c(w_{i-n+1},\cdots,w_{i-1})}, \\ \qquad\qquad 0<c(w_{i-n+1},\ldots,w_i) \end{cases} \tag{3}$$

where $\alpha(\cdot)$ is backing-off factor, $d_{c(\cdot)}$ is discount coefficient factor, and $c(\cdot)$ is word n-gram count accordingly. We note that the backing-off factor $\alpha(\cdot)$ is dependent on $d_{c(\cdot)}$. For Equation 3, we have the following assumptions:

1) Given a history $w_{i-n+1},\ldots,w_{i-1}$, the higher its occurrence, the better backing-off factor it has, and thus better probability estimate can be obtained in its backing-off condition.
2) Propability estimate from the second term is always better than that from the first term for a given n-gram.

C. Proposed probability estimate for code-switching data

In code-switching speech recognition, data sparsity issue is more severe. This is because a lot of cross-lingual n-grams do not occur frequently, and the probabilities of such n-grams are poorly estimated. Earlier work usually employs word-class n-gram language modeling approach to alleviating the data sparsity problem [26]–[28]. However, it is hard to obtain desirable word-classes using data driven based method under code-switching scenario. This is because code-switching data are much sparser than monolingual data, and hence it is difficult to discover meaningful classes. Besides, it is even harder when we try to cluster words that has similar semantic meaning but from different languages into one class due to data sparsity issue.

In this paper, we propose an improved word-class based language modeling method where only infrequent words are clustered and high frequent words are dealt with singleton classes themselves. Compared with traditional word based n-gram method, our method alleviate the data sparsity issue. This is because we use word-classes for those infrequent words, yielding increased count number of the similar events, and hence robust probability estimate for those infrequent n-grams. Moreover, compared with the normal word-class n-gram method of which word clustering is done with overall word vocabulary, the proposed "restricted" word clustering method has the advantages: a) it is much simpler to cluster words into classes since only those infrequent words are clustered, and b) it well preserves the discriminative features of the word n-gram language models for those higher frequency words. This is because those frequent words are not clustered. These can be explained with Equation 3 and corresponding assumptions. For instance, given a higher frequency word w_i, the proposed method can potentially boost its probability estimate in the case of $c(w_{i-n+1},\ldots,w_i) = 0$, since our method can make more n-gram occur satisfying assumption 1) as mentioned. On the contrary, if w_i is a infrequent word and being clustered, more probabilities of the n-gram containing the word will be estimated using the second term of Equation 3. Therefore we can obtain better n-gram probability estimate according to the assumption 2) correspondingly.

D. Perplexity results on SEAME data

In this section, we report the perplexity results of the proposed method for word-class based n-gram language modelling on SEAME data as described in Table I. To evaluate the effectiveness of the proposed method, we first build word-class based n-gram language models, then we interpolate the resulting models with the corresponding word based n-gram language models. Finally, we conduct perplexity test on the two *dev* sets in Table I. Figure 1 plots perplexity results versus count threshold below which words are clustered into 500 word-classes using Brown clustering method. Here we fix the total classes as 500 when the count threshold is changed.

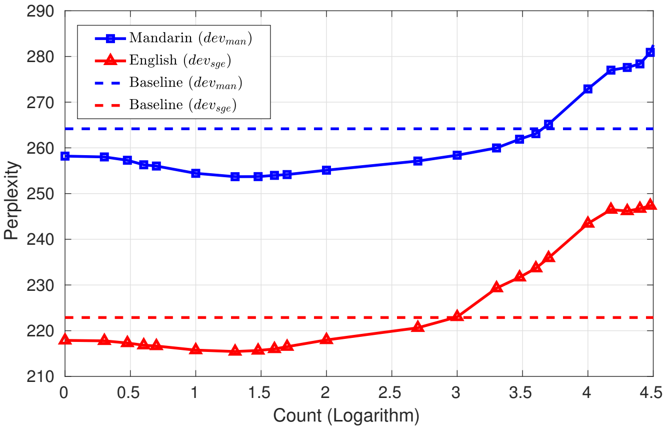

Fig. 1. Perplexity results versus word count threshold with which words that have fewer counts are clustered; here number of classes from the clustered words are fixed with 500.

From Figure 1, we can see clustering those lower frequent words always yields lower perplexity. However when those most frequent words are clustered as well, perplexity results are degraded. Actually, we also observed the similar phenomenon with different word-classes as well. This suggests those frequent words tend to be a separate classes themselves, and clustering them with other words would reduce discriminative capability of language models.

Figure 2 depicts perplexity results versus number of word-classes given a fixed word count threshold 10 to cluster those lower frequent words.

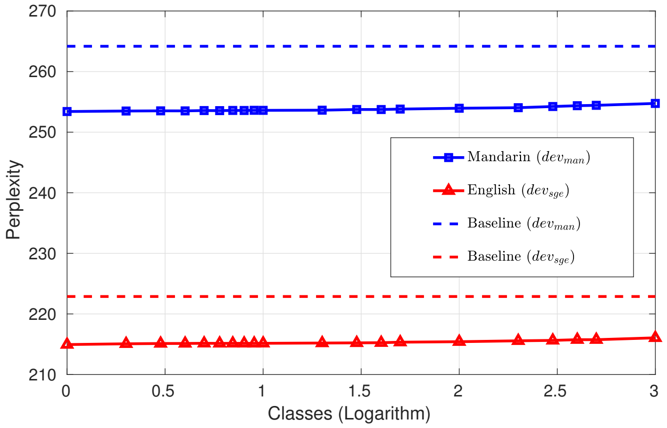

Fig. 2. Perplexity results versus number of word-classes clustered from those words whose frequency count is no more than 10.

We notice the perplexity results in Figure 2 are almost irrelevant with number of word-classes, which indicates the Brown clustering method fails to discovery meaningful word classes. The main reason should be due to data sparsity problem. Since majority of word n-gram counts are low, it is hard to exploit those word n-grams that have higher counts to cluster. Besides, in code-switching scenario, it is hard to cluster those words with similar meaning but from different languages into the same class by data driven based word cluster method. This is because those words probably appear in complete different n-gram contexts, as a result, they end up with different classes.

Table IV reports the perplexity results with the proposed word-class n-gram language modeling method (denoted as Rest. Class) on SEAME data. From Table IV, it can be seen the proposed method has consistently yielded improved perplexity results, over the baseline word n-gram language models that are trigram and trained with Kneser-Ney smoothing method. Specifically, it relatively achieves 3.69% and 3.19% perplexity reduction on dev_{man} and dev_{sge} data sets respectively.

In contrast, Table IV also reports the results from “Overall Class LM” method in which all words are clustered. The interesting thing is that the perplexity results are very close between the two word-class LMs, but the differences are remarkably enlarged when they are interpolated with word n-gram LMs respectively. The “Overall Class LM” produces degraded results as shown in Figure 1. We note that interpolating factor between the word n-gram and word-class n-gram language models is fixed with 0.6.

TABLE IV
PERPLEXITY RESULTS OF THE PROPOSED WORD-CLASS BASED N-GRAM LANGUAGE MODELS ON THE SEAME *dev* SETS

Language models	Perplexity	
	dev_{man}	dev_{sge}
Word LM	264.18	222.88
Overall Class LM	268.28	228.10
Rest. Class LM	265.66	228.65
Word LM + Overall Class LM	265.94	226.78
Word LM + Rest. Class LM	**254.43**	**215.76**

IV. PERPLEXITY RESULTS WITH THIRD-PARTY DATA

In code-switching language modelling, in-domain data is generally very limited, leading to poor estimate of language models as a result. We attempt to utilise more third-party data in Table III to boost language model performance in this section. Since out-domain code-switching data is also hard to access as well, we try to simulate a kind of code-switching data set by merging the English and Mandarin monolingual data sets as mentioned. This can be realised in terms of word-class based language modeling environment, provided words from different languages are clustered into the same class.

However, the key point is how to build word-class language models using the third-party data. We have two alternatives. First, we only use the SEAME word vocabulary to do word clustering on the third-party data set. Secondly, we only cluster those lower frequency words in the SEAME data set on the third-party data sets. Here we choose the second method. It not only mitigates the data sparsity issue of the SEAME data, but also brings smaller change to the syntactic context of those frequent words. For comparison, we also use third-party data to build word language models to interpolate with the SEAME word language models.

Table V presents the perplexity results of the proposed method with different third-party data sets employed.

TABLE V
PERPLEXITY RESULTS OF THE PROPOSED METHOD ON THE SEAME *dev* SETS, TAKING RESULTS OF THE WORD BASED LANGUAGE MODELS (LM) AS A BASELINE. THE RESULTING LMS ARE OBTAINED BY INTERPOLATING THE SEAME WORD LMS WITH CORRESPONDING WORD-CLASS LMS THAT ARE BUILT WITH SEAME DATA, MANDARIN DATA, ENGLISH DATA, AND THEIR MIXED DATA RESPECTIVELY

Data	Language models (Interpolation)	Perplexity	
		dev_{man}	dev_{sge}
Seame	Word LM	264.18	222.88
Seame	Word LM + Rest. Class LM	254.43	215.76
+Mandarin	Word LM + Word LM	260.31	230.05
+Mandarin	Word LM + Rest. Class LM	253.90	225.61
+English	Word LM + Word LM	264.12	202.54
+English	Word LM + Rest. Class LM	255.85	**186.27**
+Mixed data	Word LM + Word LM	253.16	203.33
+Mixed data	Word LM + Rest. Class LM	**249.94**	190.60

We observe from Table V that using third-party data helps in general. This is particularly true when third-party data contains both languages. As seen in the "Mixed data" case of Table V, we gain 5.39% and 14.48% relative perplexity reductions respectively in the case of using mixed third-party data. However when only monolingual (or severely biased to one of languages) data is employed it is not guaranteed to obtain improved results. As can be seen in the case of "+Mandarin" in Table V, we got even worse perplexity results on the dev_{sge} data set. This may be because when one monolingual data is used a lot of words from another language cannot be clustered reasonably, consequently resulting in poor performance on the language that has no data to cluster.

Besides in terms of effectiveness for using third-party data, results of Table V also suggest the proposed word-class language models get better results compared with corresponding word language models when both are interpolated with SEAME word language models.

For word-class language modeling ,we notice that we fix word-class number, word frequency count to cluster, and interpolating factor between word and word-class language models in Table V. They are 500, 10 and 0.6 respectively. For word language models from the third-party data, they are all Kneser-Ney smoothed, and when they are interpolated with SEAME word language models, the corresponding interpolating factors are optimized.

V. Word-class Language model rescoring

To test the effectiveness of the proposed language modeling method, we perform lattice rescoring in this Section.

A. Speech recognition System

Our acoustic models are trained with lattice-free Maximum Mutual Information (LF-MMI) [29] criteria, on the top of a 7-layer Time Delay Neural Network (TDNN) similar to one proposed in [30] but using Rectified Linear Units (ReLU) as neurons instead. The front-ends are made of 40-dimensional MFCC plus 100-dimensional i-vectors [31]. They are LDA transformed over ± 1 feature window before fed to the TDNN as input. The output TDNN has 5950 nodes, which are the HMM senones. Except for acoustic models as mentioned above, the lexicon and language models are all built with training transcripts as well.

B. N-best rescoring

In our work, we employ a similified language model rescoring method. The rescoring is performed on the n-best hypthoses instead of lattice. To do this, it is necessary to generate lattice. In our experiments, all lattices are generated with word-based trigram language models. We then generate n-best hypotheses from lattice. and use various language models to re-rank the n-best combined with corresponding acoustic scores. Table VI reports the language model rescoring results. From Table VI, we get rather small but consistent WER performance improvement using the proposed word-class based language modeling method. The marginal improvements implicitly say the contribution of the n-gram language modeling methods is rather limited due to strong acoustic modeling method employed.

TABLE VI
WER results of Language Model (LM) rescoring, by reranking 10-best hypotheses from lattice

Setup	dev_{man} (%)	dev_{sge} (%)
Oracle	16.90	24.36
Word LM	25.74	37.27
Word LM + Rest. Class LM	25.65	37.14

VI. Conclusions

In this paper we proposed an improved word-class based language modeling method for code-switching speech recognition task, in which only low-frequency words are clustered, while those high-frequency words are dealt with singleton classes themselves. We first demonstrated its effectiveness with regard to perplexity reduction of language models. Compared with word n-gram language models, we achieved up to 3.69% perplexity reduction when only SEAME transcript data is employed; and 14.48% reduction when more third-party data is introduced. We then tried language model rescoring in speech recognition task, however, we got very marginal but consistent WER reduction, using state-of-the-art TDNN LF-MMI acoustic modeling method.

REFERENCES

[1] 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 {M}andarin-{E}nglish code-switching conversational speech," in *Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2012.

[2] N. T. Vu, H. Adel, and T. Schultz, "An investigation of code-switching attitude dependent language modeling," in *International Conference on Statistical Language and Speech Processing*. Springer, 2013, pp. 297–308.

[3] C.-H. Wu, H.-P. Shen, and Y.-T. Yang, "Chinese-English phone set construction for code-switching ASR using acoustic and DNN-extracted articulatory features," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 858–862, 2014.

[4] C.-F. Yeh, Y.-C. Lin, and L.-S. Lee, "Minimum phone error model training on merged acoustic units for transcribing bilingual code-switched speech," in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*. IEEE, 2012, pp. 320–324.

[5] C.-F. Yeh and L.-S. Lee, "Transcribing code-switched bilingual lextures using deep neural networks with unit merging in acoustic modeling," in *Proc. of IEEE International Conference On Acoustic, Speech and Signal Processing (ICASSP)*, 2014.

[6] ——, "An improved framework for recognizing highly imbalanced bilingual code-switched lectures with cross-language acoustic modeling and frame-level language identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 7, pp. 1144–1159, 2015.

[7] E. Yilmaz, H. V. D. Heuvel, and D. V. Leeuwen, "Investigating bilingual deep neural networks for automatic recognition of code-switching {F}risian speech," in *Procedia Computer Science*, 2016.

[8] D. Janiszek, R. D. Mori, and F. Bechet, "Data augmentation and language model adaptation," *Proceedings 2001 International Conference Acoustic Speech and Signal Process*, 2001.

[9] M. Creutz, S. Virpioja, and A. Kovaleva, "Web augmentation of language models for continuous speech recognition of SMS text messages," *Computational Linguistics*, no. April, pp. 157–165, 2009.

[10] J. T. Goodman, "Putting it all together: language model combination," vol. 3, pp. 1647–1650, 2000.

[11] X. Liu, M. Gales, J. Hieronymus, and PC, "Language model combination and adaptation using weighted finite state transducers," *Speech and Signal*, pp. 5390–5393, 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5494941

[12] Y. Li and P. Fung, "Code-Switch Language Model with Inversion Constraints for Mixed Language Speech Recognition." in *COLING*, 2012, pp. 1671–1680.

[13] ——, "Code switch language modeling with functional head constraint," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4913–4917.

[14] H. Adel, K. Kirchhoff, D. Telaar, N. T. Vu, T. Schlippe, and T. Schultz, "Features for factored language models for code-Switching speech." in *SLTU*, 2014, pp. 32–38.

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

[16] F. AlGhamdi, G. Molina, M. Diab, T. Solorio, A. Hawwari, V. Soto, and J. Hirschberg, "Part of speech tagging for code switched data," *EMNLP 2016*, p. 98, 2016.

[17] Ö. Çetinoğlu, S. Schulz, and N. T. Vu, "Challenges of computational processing of code-switching," *arXiv preprint arXiv:1610.02213*, 2016.

[18] H. Adel, N. T. Vu, F. Kraus, T. Schlippe, H. Li, and T. Schultz, "Recurrent neural network language modeling for code switching conversational speech," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8411–8415.

[19] H. Adel, N. T. Vu, and T. Schultz, "Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling." in *ACL (2)*, 2013, pp. 206–211.

[20] 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 *INTERSPEECH*, 2014, pp. 1415–1419.

[21] D.-C. Lyu, T. P. Tan, E. Chng, and H. Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia." in *INTERSPEECH*, vol. 10, 2010, pp. 1986–1989.

[22] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 334–342.

[23] V. Kuznetsov, H. Liao, M. Mohri, M. Riley, and B. Roark, "Learning n-gram language models from uncertain data," *Proceedings of Interspeech (to appear)*, 2016.

[24] H. Zhang and D. Chiang, "Kneser-Ney Smoothing on Expected Counts." in *ACL (1)*, 2014, pp. 765–774.

[25] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, and Others, "The HTK book," *Cambridge university engineering department*, vol. 3, p. 175, 2002.

[26] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[27] G. Moore and S. J. Young, "Class-based language model adaptation using mixtures of word-class weights." in *INTERSPEECH*, 2000, pp. 512–515.

[28] R. Botros, K. Irie, M. Sundermeyer, and H. Ney, "On efficient training of word classes and their application to recurrent neural network language models." in *INTERSPEECH*, 2015, pp. 1443–1447.

[29] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for {ASR} based on lattice-free {MMI}," in *Proc. of INTERSPEECH*, 2016.

[30] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. of INTERSPEECH*, 2015.

[31] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013.