Exploration of the Impact of Maximum Entropy in Recurrent Neural Network Language Models for Code-Switching Speech

Ngoc Thang Vu[1,2] **and Tanja Schultz**[1]
[1]Karlsruhe Institute of Technology (KIT), [2]University of Munich (LMU), Germany
thangvu@cis.lmu.de, tanja.schultz@kit.edu

Abstract

This paper presents our latest investigations of the jointly trained maximum entropy and recurrent neural network language models for Code-Switching speech. First, we explore extensively the integration of part-of-speech tags and language identifier information in recurrent neural network language models for Code-Switching. Second, the importance of the maximum entropy model is demonstrated along with a various of experimental results. Finally, we propose to adapt the recurrent neural network language model to different Code-Switching behaviors and use them to generate artificial Code-Switching text data.

1 Introduction

The term Code-Switching (CS) denotes speech which contains more than one language. Speakers switch their language while they are talking. This phenomenon appears very often in multilingual communities, such as in India, Hong Kong or Singapore. Furthermore, it increasingly occurs in former monolingual cultures due to the strong growth of globalization. In many contexts and domains, speakers switch more often between their native language and English within their utterances than in the past. This is a challenge for speech recognition systems which are typically monolingual. While there have been promising approaches to handle Code-Switching in the field of acoustic modeling, language modeling is still a great challenge. The main reason is a shortage of training data. Whereas about 50h of training data might be sufficient for the estimation of acoustic models, the transcriptions of these data are not enough to build reliable language models. In this paper, we focus on exploring and improving the language model for Code-switching speech and as a result improve the automatic speech recognition (ASR) system on Code-Switching speech.

The main contribution of the paper is the extensive investigation of jointly trained maximum entropy (ME) and recurrent neural language models (RNN LMs) for Code-Switching speech. We revisit the integration of part-of-speech (POS) tags and language identifier (LID) information in recurrent neural network language models and the impact of maximum entropy on the language model performance. As follow-up to our previous work in (Adel, Vu et al., 2013), here we investigate whether a recurrent neural network alone without using ME is a suitable model for Code-Switching speech. Afterwards, to directly use the RNN LM in the decoding process of an ASR system, we convert the RNN LM into the n-gram language model using the text generation approach (Deoras et al., 2011; Adel et al., 2014); Furthermore motivated by the fact that Code-Switching is speaker dependent (Auer, 1999b; Vu et al., 2013), we first adapt the recurrent neural network language model to different Code-Switching behaviors and then generate artificial Code-Switching text data. This allows us to train an accurate n-gram model which can be used directly during decoding to improve ASR performance.

The paper is organized as follows: Section 2 gives a short overview of related works. In Section 3, we describe the jointly trained maximum entropy and recurrent neural network language models and their extension for Code-Switching speech. Section 4 gives a short description of the SEAME corpus. In Section 5, we summarize the most important experiments and results. The study is concluded in Section 6 with a summary.

2 Related Work

This section gives a brief introduction about the related research regarding Code-Switching and re-

Proceedings of The First Workshop on Computational Approaches to Code Switching, pages 34–41,
October 25, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

current language models.
In (Muysken, 2000; Poplack, 1978; Bokamba, 1989), the authors observed that code switches occur at positions in an utterance following syntactical rules of the involved languages. Code-Switching can be regarded as a speaker dependent phenomenon (Auer, 1999b; Vu et al., 2013). However, several particular Code-Switching patterns are shared across speakers (Poplack, 1980). Furthermore, part-of-speech tags might be useful features to predict Code-Switching points. The authors of (Solorio et al., 2008b; Solorio et al., 2008a) investigate several linguistic features, such as word form, LID, POS tags or the position of the word relative to the phrase for Code-Switching prediction. Their best result is obtained by combining all those features. (Chan et al., 2006) compare four different kinds of n-gram language models to predict Code-Switching. They discover that clustering all foreign words into their POS classes leads to the best performance. In (Li et al., 2012; Li et al., 2013), the authors propose to integrate the equivalence constraint into language modeling for Mandarin and English Code-Switching speech recorded in Hong Kong.

In the last years, neural networks have been used for a variety of tasks, including language modeling (Mikolov et al., 2010). Recurrent neural networks are able to handle long-term contexts since the input vector does not only contain the current word but also the previous hidden layer. It is shown that these networks outperform traditional language models, such as n-grams which only contain very limited histories. In (Mikolov et al., 2011a), the network is extended by factorizing the output layer into classes to accelerate the training and testing processes. The input layer can be augmented to model features, such as POS tags (Shi et al., 2011; Adel, Vu et al., 2013). Furthermore, artificial text can be automatically generated using recurrent neural networks to enlarge the amount of training data (Deoras et al., 2011; Adel et al., 2014).

3 Joint maximum entropy and recurrent neural networks language models for Code-Switching

3.1 Recurrent neural network language models

The idea of RNN LMs is illustrated in Figure 1. Vector $w(t)$ forms the input of the recurrent neural network. It represents the current word using 1-of-N coding. Thus, its dimension equals the size of the vocabulary. Vector $s(t)$ contains the state of the network - 'hidden layer'. The network is trained using back-propagation through time (BPTT), an extension of the back-propagation algorithm for recurrent neural networks. With BPTT, the error is propagated through recurrent connections back in time for a specific number of time steps t. Hence, the network is able to capture a longer history than a traditional n-gram LM. The matrices U, V and W contain the weights for the connections between the layers. These weights are learned during the training phase.

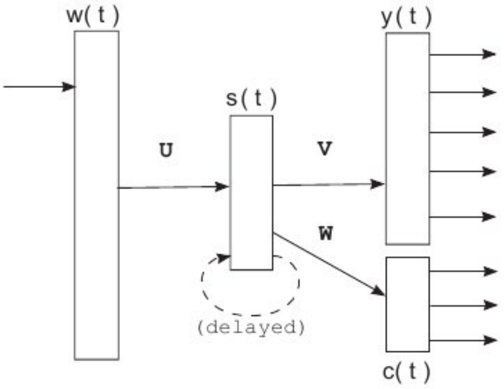

Figure 1: RNN language model

To accelerate the training process, (Mikolov et al., 2011a) factorized the output layer into classes based on simple frequency binning. Every word belongs to exactly one class. Vector $c(t)$ contains the probabilities for each class and vector $w(t)$ provides the probabilities for each word given its class. Hence, the probability $P(w_i|history)$ is computed as shown in equation 1.

$$P(w_i|history) = P(c_i|s(t))P(w_i|c_i, s(t)) \quad (1)$$

Furthermore in (Mikolov et al., 2011b), the authors proposed to jointly train the RNN with ME - *RMM-ME* - to improve the language model and also ASR performance. The ME can be seen as a weight matrix which directly connects the input with the output layer as well as the input with the class layer. This weight matrix can be trained jointly with the recurrent neural network. “Direct-order” and “direct connection” are the two important parameters which define the length of history and the number of the trained connections.

3.2 Code-Switching language models

To adapt RNN LMs to the Code-Switching task, (Adel, Vu et al., 2013) analyzed the SEAME corpus and observed that there are words and POS tags which might have a high potential to predict Code-Switching points. Therefore, it has been proposed to integrate the POS and LID information into the RNN LM. The idea is to factorize the output layer into classes which provide language information. By doing that, it is intended to not only predict the next word but also the next language. Hence according to equation 1, the probability of the next language is computed first and then the probability of each word given the language. In that work, four classes were used: English, Mandarin, other languages and particles. Moreover, a vector $f(t)$ which contains the POS information is added to the input layer. This vector provides the corresponding POS of the current word. Thus, not only the current word is activated but also its features. Since the POS tags are integrated into the input layer, they are also propagated into the hidden layer and back-propagated into its history $s(t)$. Hence, not only the previous features are stored in the history but also features from several time steps in the past.

In addition to that previous work, the experiments in this paper aim to explore the source of the improvements observed in (Adel, Vu et al., 2013). We now clearly distinguish between the impacts due to the long but unordered history of the RNN and the effects of the maximum entropy model which also captures information about the most recent word and POS tag in the history.

4 SEAME corpus

To conduct research on Code-Switching speech we use the SEAME corpus (South East Asia Mandarin-English). It is a conversational Mandarin-English Code-Switching speech corpus recorded by (D.C. Lyu et al., 2011). Originally, it was used for the research project "Code-Switch" which was jointly performed by Nanyang Technological University (NTU) and Karlsruhe Institute of Technology (KIT) from 2009 until 2012. The corpus contains 63 hours of audio data which has been recorded and manually transcribed in Singapore and Malaysia. The recordings consist of spontaneously spoken interviews and conversations. The words can be divided into four language categories: English words (34.3% of all tokens), Mandarin words (58.6%), particles (Singaporean and Malayan discourse particles, 6.8% of all tokens) and others (other languages, 0.4% of all tokens). In total, the corpus contains 9,210 unique English and 7,471 unique Mandarin words. The Mandarin character sequences have been segmented into words manually. The language distribution shows that the corpus does not contain a clearly predominant language. Furthermore, the number of Code-Switching points is quite high: On average, there are 2.6 switches between Mandarin and English per utterance. Additionally, the duration of the monolingual segments is rather short: More than 82% of the English segments and 73% of the Mandarin segments last less than one second. The average duration of English and Mandarin segments is only 0.67 seconds and 0.81 seconds, respectively. This corresponds to an average length of monolingual segments of 1.8 words in English and 3.6 words in Mandarin.

For the task of language modeling and speech recognition, the corpus has been divided into three disjoint sets: training, development and evaluation set. The data is assigned to the three different sets based on the following criteria: a balanced distribution of gender, speaking style, ratio of Singaporean and Malaysian speakers, ratio of the four language categories, and the duration in each set. Table 1 lists the statistics of the SEAME corpus.

	Training	Dev	Eval
# Speakers	139	8	8
Duration(hours)	59.2	2.1	1.5
# Utterances	48,040	1,943	1,029
# Words	575,641	23,293	11,541

Table 1: Statistics of the SEAME corpus

5 Experiments and Results

This section presents all the experiments and results regarding language models and ASR on the development and the evaluation set of the SEAME corpus. However, the parameters were tuned only on the development set.

5.1 LM experiments

5.1.1 Baseline n-gram

The n-gram language model served as the baseline in this work. We used the SRI language model toolkit (Stolcke, 2002) to build the CS 3-gram baseline from the SEAME training transcriptions containing all words of the transcriptions. Modified Kneser-Ney smoothing (Rosenfeld, 2000) was applied. In total, the vocabulary size is around 16k words. The perplexities (PPLs) are 268.4 and 282.9 on the development and evaluation set respectively.

5.1.2 Exploration of ME and of the integration of POS and LID in RNN

To investigate the effect of POS and LID integration into the RNN LM and the importance of the ME, different RNN LMs were trained.

The first experiment aims at investigating the importance of using LID information for output layer factorization. All the results are summarized in table 2. The first RNNLM was trained with a hidden layer of 50 nodes and without using output factorization and ME. The PPLs were 250.8 and 301.1 on the development and evaluation set, respectively. We observed some gains in terms of PPL on the development set but not on the evaluation set compared to the n-gram LM. Even using ME and factorizing the output layer into four classes based on frequency binning (fb), the same trend could be noticed - only the PPL on the development set was improved. Four classes were used to have a fair comparison with the output factorization with LID. However after including the LID information into the output layer, the PPLs were improved on both data sets. On top of that, using ME provides some additional gains. The results indicate that LID is a useful information source for the Code-Switching task. Furthermore, the improvements are independent of the application of ME.

Model	Dev	Eval
CS 3-gram	268.4	282.9
RNN LM	250.8	301.1
RNN-ME LM	246.6	287.9
RNN LM with fb	246.0	287.3
RNN-ME LM with fb	256.0	294.0
RNN LM with LID	241.5	274.4
RNN-ME LM with LID	**237.9**	**269.3**

Table 2: Effect of output layer factorization

In the second experiment we investigated the use of POS information and the effect of the ME. The results in Table 3 show that an integration of POS without ME did not give any further improvement compared to RNN LM. The reason could lie in the fact that a RNN can capture a long history but not the information of the word order. Note that in the syntactic context, the word order is one of the most important information. However using ME allows using the POS of the previous time step to predict the next language and also the next word, the PPL was improved significantly on development and evaluation set. These results reveal that POS is a reasonable trigger event which can be used to support Code-Switching prediction.

Model	Dev	Eval
CS 3-gram	268.4	282.9
RNN LM	250.8	301.1
RNN-ME LM	246.6	287.9
RNN LM with POS	250.6	298.3
RNN-ME LM with POS	**233.5**	**268.0**

Table 3: Effect of ME on the POS integration into the input layer

Finally, we trained an LM by integrating the POS tags and factorizing the output layer with LID information. Again without applying ME, we observed that POS information is not helpful to improve the RNN LM. Using the ME provides a big gain in terms of PPL on both data sets. We obtained a PPL of 219.8 and 239.2 on the development and evaluation set respectively.

Model	Dev	Eval
CS 3-gram	268.4	282.9
RNN LM	250.8	301.1
RNN-ME LM	246.6	287.9
RNN LM with POS + LID	243.9	277.1
RNN-ME LM with POS+ LID	**219.8**	**239.2**

Table 4: Effect of ME on the integration of POS and the output layer factorization using LID

5.1.3 Training parameters

Moreover, we investigated the effect of different parameters, such as the backpropagation through time (BPTT) step, the direct connection order and the amount of direct connections on the performance of the RNN-ME LMs. Therefore, different LMs were trained with varying values for these parameters. For each parameter change, the remaining parameters were fixed to the most suitable value which has been found so far.

First, we varied the BPTT step from 1 to 5. The BPTT step defines the length of the history which is incorporated to update the weight matrix of the

RNN. The larger the BPTT step is, the longer is the history which is used for learning. Table 5 shows the perplexities on the SEAME development and evaluation sets with different BPTT steps. The results indicate that increasing BPTT might improve the PPL. The best PPL can be obtained with a BPTT step of 4. The big loss in terms of PPL by using a BPTT step of 5 indicates that too long histories might hurt the language model performance. Another reason might be the limitation of the training data.

BPTT	1	2	3	4	5
Dev	244.7	224.6	222.8	**219.8**	266.8
Eval	281.1	241.4	242.8	**239.2**	284.5

Table 5: Effect of the BPTT step

It has been shown in the previous section, that ME is very important to improve the PPL especially for the Code-Switching task, we also trained several RNN-ME LMs with various values for "direct order" and "direct connection". Table 6 and 7 summarize the PPL on the SEAME development and evaluation set. The results reveal that the larger the direct order is, the lower is the PPL. We observed consistent PPL improvement by increasing the direct order. However, the gain seems to be saturated after a direct order of 3 or 4. In this paper, we choose to use a direct order of 4 to train the final model.

Direct order	1	2	3	4
Dev	238.6	231.7	220.5	**219.8**
Eval	271.8	261.4	240.7	**239.2**

Table 6: Effect of the direct order

Since the "direct order" is related to the length of the context, the size of the "direct connection" is a trade off between the size of the language model and also the amount of the training data. Higher "direct connection" leads to a larger model and might improve the PPL if the amount of training data is enough to train all the direct connection weights. The results with four different data points (50M, 100M, 150M and 200M) show that the best model can be obtained on SEAME data set by using 100M of direct connection.

5.1.4 Artificial Code-Switching text generation using RNN

The RNN LM demonstrates a great improvement over the traditional n-gram language model. However, it is inefficient to use the RNN LM directly in the decoding process of an ASR system. In order to convert the RNN into a n-gram language model, a text generation method which was proposed in (Deoras et al., 2011) can be applied. Moreover, it allows to generate more training data which might be useful to improve the data sparsity of the language modeling task for Code-Switching speech. In (Deoras et al., 2011), the authors applied the Gibb sampling method to generate artificial text based on the probability distribution provided by the RNNs. We applied that technique in (Adel et al., 2014) to generate Code-Switching data and were able to improve the PPL and ASR performance on CS speech. In addition to that previous work, we now propose to use several Code-Switching attitude dependent language models instead of the final best RNN LM.

#Connection	50M	100M	150M	200M
Dev	226.2	**219.8**	224.7	224.6
Eval	244.7	**239.2**	243.7	242.0

Table 7: Effect of the number of direct connections

Code-Switching attitude dependent language modeling Since POS tags might have a potential to predict Code-Switch points, (Vu et al., 2013) performed an analysis of these trigger POS tags on a speaker level. The CS rate for each tag was computed for each speaker. Afterwards, we calculated the minimum, maximum and mean values as well as standard deviations. We observed that the spread between minimum and maximum values is quite high for most of the tags. It indicates that although POS information may trigger a CS event, it is rather speaker dependent.

Motivated by this observation, we performed k-mean clustering of the training text into three different portions of text data which describe different Code-Switching behaviors (Vu et al., 2013). Afterwards, the LM was adapted with each text portion to obtain Code-Switching attitude dependent language models. By using these models, we could improve both PPL and ASR performance for each speaker.

Artificial text generation To generate artificial text, we first adapted the best RNN-ME LM described in the previous section to three different Code-Switching attitudes. Afterwards, we generated three different text corpora based on these specific Code-Switching attitudes. Each corpus contains 100M tokens. We applied the SRILM toolkit (Stolcke, 2002) to train n-gram language model and interpolated them linearly with the weight $= \frac{1}{3}$. Table 8 shows the perplexity of the resulting n-gram models on the SEAME development and evaluation set. To make a comparison, we also used the unadapted best RNN-ME LM to generate two different texts, one with 300M tokens and another one with 235M tokens (Adel et al., 2014). The results show that the n-gram LMs trained with only the artificial text data can not outperform the baseline CS 3-gram. However they provide some complementary information to the baseline CS 3-gram LM. Therefore, when we interpolated them with the baseline CS 3-gram, the PPL was improved all the cases. Furthermore by using the Code-Switching attitude dependent language models to generate artificial CS text data, the PPL was slightly improved compared to using the unadapted one. The final 3-gram model (*Final 3-gram*) was built by interpolating all the Code-Switching attitude dependent 3-gram and the baseline CS 3-gram. It has a PPL of 249.3 and 266.9 on the development set and evaluation set.

Models	Dev	Eval
CS 3-gram	268.4	282.9
300M words text	391.3	459.5
+ CS 3-gram	250.0	270.9
235M words text	385.1	454.6
+ CS 3-gram	249.5	270.5
100M words text I	425.4	514.4
+ CS 3-gram	251.4	274.5
100M words text II	391.8	421.6
+ CS 3-gram	251.6	266.4
100M words text III	390.3	428.1
+ CS 3-gram	250.6	266.9
Interpolation of I, II and III	377.5	416.1
+ CS 3-gram (Final n-gram)	**249.3**	**266.9**
RNN-ME LM + POS + LID	**219.8**	**239.2**

Table 8: PPL of the N-gram models trained with artificial text data

5.2 ASR experiments

For the ASR experiments, we applied BioKIT, a dynamic one-pass decoder (Telaar et al., 2014). The acoustic model is speaker independent and has been trained with all the training data. To extract the features, we first trained a multilayer perceptron (MLP) with a small hidden layer with 40 nodes. The output of this hidden layer is called *bottle neck features* and is used to train the acoustic model. The MLP has been initialized with a multilingual multilayer perceptron as described in (Vu et al., 2012). The phone set contains English and Mandarin phones, filler models for continuous speech (+noise+, +breath+, +laugh+) and an additional phone +particle+ for Singaporean and Malayan particles. The acoustic model applied a fully-continuous 3-state left-to-right HMM. The emission probabilities were modeled with Gaussian mixture models. We used a context dependent acoustic model with 3,500 quintphones. Merge-and-split training was applied followed by six iterations of Viterbi training. To obtain a dictionary, the CMU English (CMU Dictionary, 2014) and Mandarin (Hsiao et al., 2008) pronunciation dictionaries were merged into one bilingual pronunciation dictionary. Additionally, several rules from (Chen et al., 2010) were applied which generate pronunciation variants for Singaporean English.

As a performance measure for decoding Code-Switching speech, we used the mixed error rate (MER) which applies word error rates to English and character error rates to Mandarin segments (Vu et al., 2012). With character error rates for Mandarin, the performance can be compared across different word segmentations. Table 9 shows the results of the baseline CS 3-gram LM, the 3-gram LM trained with 235M artificial words interpolated with CS 3-gram LM and the final 3-gram LM described in the previous section. Compared to the baseline system, we are able to improve the MER by up to 3% relative. Furthermore, a very small gain can be observed by using the Code-Switching attitude dependent language model compared to the unadapted best RNN-ME LM.

Model	Dev	Eval
CS 3-gram	40.0%	34.3%
235M words text + CS-3gram	39.4%	33.4%
Final 3-gram	**39.2%**	**33.3%**

Table 9: ASR results on SEAME data

6 Conclusion

This paper presents an extensive investigation of the impact of maximum entropy in recurrent neural network language models for Code-Switching speech. The experimental results reveal that factorization of the output layer of the RNN using LID always improved the PPL independent whether the ME is used. However, the integration of the POS tags into the input layer only improved the PPL in combination with ME. The best LM can be obtained by jointly training the ME and the RNN LM with POS integration and factorization using LID. Moreover, using the RNN-ME LM allows generating artificial CS text data and therefore training an n-gram LM which carries the information of the RNN-ME LM. This can be directly used during decoding to improve ASR performance on Code-Switching speech. On the SEAME development and evaluation set, we obtained an improvement of up to 18% relative in terms of PPL and 3% relative in terms of MER.

7 Acknowledgment

This follow-up work on exploring the impact of maximum entropy in recurrent neural network language models for Code-Switching speech was motivated by the very useful comments and suggestions of the SLSP reviewers, for which we are very grateful.

References

H. Adel, N.T. Vu, F. Kraus, T. Schlippe, and T. Schultz. *Recurrent Neural Network Language Modeling for Code Switching Conversational Speech* In: Proceedings of ICASSP 2013.

H. Adel, K. Kirchhoff, N.T. Vu, D.Telaar, T. Schultz *Comparing Approaches to Convert Recurrent Neural Networks into Backoff Language Models For Efficient Decoding* In: Proceedings of Interspeech 2014.

P. Auer *Code-Switching in Conversation* Routledge 1999.

P. Auer *From codeswitching via language mixing to fused lects toward a dynamic typology of bilingual speech* In: International Journal of Bilingualism, vol. 3, no. 4, pp. 309-332, 1999.

E.G. Bokamba *Are there syntactic constraints on code-mixing?* In: World Englishes, vol. 8, no. 3, pp. 277-292, 1989.

J.Y.C. Chan, PC Ching, T. Lee, and H. Cao *Automatic speech recognition of Cantonese-English code-mixing utterances* In: Proceeding of Interspeech 2006.

W. Chen, Y. Tan, E. Chng, H. Li *The development of a Singapore English call resource* In: Proceedings of Oriental COCOSDA, 2010.

Carnegie Mellon University *CMU pronouncation dictionary for English* Online: http://www.speech.cs.cmu.edu/cgi-bin/cmudict, retrieved in July 2014

D.C. Lyu, T.P. Tan, E.S. Cheng, H. Li *An Analysis of Mandarin-English Code-Switching Speech Corpus: SEAME* In: Proceedings of Interspeech 2011.

A. Deoras, T. Mikolov, S. Kombrink, M. Karafiat, S. Khudanpur *Variational approximation of long-span language models for LVCSR* In: Proceedings of ICASSP 2011.

R. Hsiao, M. Fuhs, Y. Tam, Q. Jin, T. Schultz *The CMU-InterACT 2008 Mandarin transcription system* In: Procceedings of ICASSP 2008.

Y. Li, P. Fung *Code-Switch Language Model with Inversion Constraints for Mixed Language Speech Recognition* In: Proceedings of COLING 2012.

Y. Li, P. Fung *Improved mixed language speech recognition using asymmetric acoustic model and language model with Code-Switch inversion constraints* In: Proceedings of ICASSP 2013.

M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. *Building a large annotated corpus of english: The penn treebank* In: Computational Linguistics, vol. 19, no. 2, pp. 313-330, 1993.

T. Mikolov, M. Karafiat, L. Burget, J. Jernocky and S. Khudanpur. *Recurrent Neural Network based Language Model* In: Proceedings of Interspeech 2010.

T. Mikolov, S. Kombrink, L. Burget, J. Jernocky and S. Khudanpur. *Extensions of Recurrent Neural Network Language Model* In: Proceedings of ICASSP 2011.

T. Mikolov, A. Deoras, D. Povey, L. Burget, J.H. Cernocky *Strategies for Training Large Scale Neural Network Language Models* In: Proceedings of ASRU 2011.

P. Muysken *Bilingual speech: A typology of code-mixing* In: Cambridge University Press, vol. 11.

S. Poplack *Syntactic structure and social function of code-switching* , Centro de Estudios Puertorriquenos, City University of New York.

S. Poplack *Sometimes i'll start a sentence in spanish y termino en espanol: toward a typology of code-switching* In: Linguistics, vol. 18, no. 7-8, pp. 581-618.

D. Povey, A. Ghoshal, et al. *The Kaldi speech recognition toolkit* In: Proceedings of ASRU 2011.

R. Rosenfeld *Two decades of statistical language modeling: Where do we go from here?* In: Proceedings of the IEEE 88.8 (2000): 1270-1278.

T. Schultz, P. Fung, and C. Burgmer, Detecting code-switch events based on textual features.

Y. Shi, P. Wiggers, M. Jonker *Towards Recurrent Neural Network Language Model with Linguistics and Contextual Features* In: Proceedings of Interspeech 2011.

T. Solorio, Y. Liu *Part-of-speech tagging for English-Spanish code-switched text* In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008.

T. Solorio, Y. Liu *Learning to predict code-switching points* In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008.

A. Stolcke *SRILM-an extensible language modeling toolkit.* In: Proceedings of Interspeech 2012.

D. Telaar, et al. *BioKIT - Real-time Decoder For Biosignal Processing* In: Proceedings of Interspeech 2014.

N.T. Vu, D.C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.S. Chng, T. Schultz, H. Li *A First Speech Recognition System For Mandarin-English Code-Switch Conversational Speech* In: Proceedings of Interspeech 2012.

N.T. Vu, H. Adel, T. Schultz *An Investigation of Code-Switching Attitude Dependent Language Modeling* In: In Statistical Language and Speech Processing, First International Conference, 2013.

N.T. Vu, F. Metze, T. Schultz *Multilingual bottleneck features and its application for under-resourced languages* In: Proceedings of SLTU, 2012.