Speaker Adaptation Using the I-Vector Technique for Bottleneck Features

Patrick Cardinal[1,2], *Najim Dehak*[1], *Yu Zhang*[1], *James Glass*[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory,
Cambridge, Massachusetts 02139, USA
[2]École de Technologie Supérieure,
Montréal, Canada

patrick.cardinal@etsmtl.ca, najim@csail.mit.edu
yzhang87@mit.edu, glass@mit.edu

Abstract

Deep Neural Networks (DNN) have been largely used and successfully applied in the context of speaker independent Automatic Speech Recognition (ASR). However, these models are not easily adapted to model a specific speaker characteristic. Recently, one approach was proposed to address this issue, which consists of using the I-vector representation as input to the DNN. The I-vector is playing the role of providing information about the speaker as well as the environmental conditions for a given recording. This approach achieved a significant improvement in the context of a hybrid system of DNN combined with Hidden Markov Model (HMM). In this paper, we study the effect of speaker adaptation based on the I-vector framework in the context of stacked bottleneck features. These features, extracted from a second level of DNNs, are modelled by a classical Gaussian Mixture Model (GMM) ASR system. The proposed approach achieved an absolute WER improvement of 1.2% on an Arabic Broadcast news task.

Index Terms: DNN, I-Vector, Bottleneck Features, Speech Recognition

1. Introduction

Over the past few years, deep neural networks (DNNs) have become increasingly popular for automatic speech recognition (ASR), due to the improvement in accuracy over conventional Gaussian mixture model (GMM) based systems. There are two main ways to use a DNN for acoustic modelling. With the *hybrid* approach, the DNN is trained to directly estimate the posteriors of hidden Markov model (HMM) states. Alternatively, with the *tandem* approach, the DNN is used to learn an effective feature representation for a particular task. The features are obtained by training a DNN to predict a phonetic target [1] such as the HMM state posteriors. The outputs of a low-dimensional internal layer within the DNN, called the *bottleneck* (BN) layer, are then used as features in the conventional HMM-GMM framework.

Many studies have been conducted with BN features in order to reduce the accuracy gap between hybrid and BN-based tandem ASR systems [2, 3, 4, 5, 6]. In the work described in [7], the authors describe a hierarchical architecture in which a second DNN is used to correct the posterior outputs (estimation of HMM state emission probabilities) by using a different set of features. They also implement low-rank matrix factorization by using a linear activation function for the BN layer. In this configuration, the BN layer was found to be most effective in the last hidden layer just prior to the output layer. This structure is also used in this work to implement BN features. Another use of BN features is to use them in hybrid systems. This idea has been successfully explored in [8, 9]. This approach will also be considered in this work.

A major drawback of DNNs is the difficulty of adapting them to a specific speaker. Most techniques developed for adapting GMMs cannot be applied to DNNs. One exception is fMLLR adaptation, a widely used technique for speaker adaptation proposed by Gales [10]. Yu *et al.* proposed the Kullback-Leibler divergence (KLD) regularization for adapting a DNN. They reported an error reduction up to 30% [11]. Recently, Saon *et al.* [12] proposed the use of I-vectors for adapting a DNN to a specific speaker. The idea is to concatenate a speaker-specific I-vector with conventional frame-based features. With the additional I-vector information, the DNN is able to learn speaker-specific differences. We successfully used this technique for the task of Arabic broadcast news transcription [13].

In this paper, we explore the use of the I-vector for speaker adaptation of DNNs used for BN feature extraction. The rest of the paper is organized as follows. Section 2 and Section 3 describe the approach used in this work to extract BN features, and how I-vectors are extracted. Section 4 describes our recent experiments and results. The paper is concluded in Section 5.

2. Bottleneck Feature Extraction

BN feature extraction typically is accomplished by using a DNN to extract discriminative features from a speech corpus. BN features can then be modelled with a GMM, allowing the use of all optimization techniques previously developed for the classical GMM based ASR system.

The BN feature extractor used in this work is a hierarchical approach described in [14], combined with a low-rank matrix factorization proposed by Sainath *et al.*[15]. This approach, low-rank stacked BN (LrSBN) has produced good results on low-resource ASR systems [7]. The LrSBN approach, summarized in Figure 1, is used to extract features for a GMM system. The input of the first layer consists of 23 critical-band energies that are obtained from a Mel filter-bank. Pitch and voicing probability are then added. 11 consecutive frames are then stacked together. Each of the 23+2 dimensions is then multiplied by a Hamming window across time, and a discrete cosine transform (DCT) is applied for dimensionality reduction. The 0th to 5th DCT coefficients are retained, resulting in a feature of dimensionality (23 + 2) x 6 = 150.

The second layer is used for correcting the posterior outputs of the first layer. In this architecture, the input features of

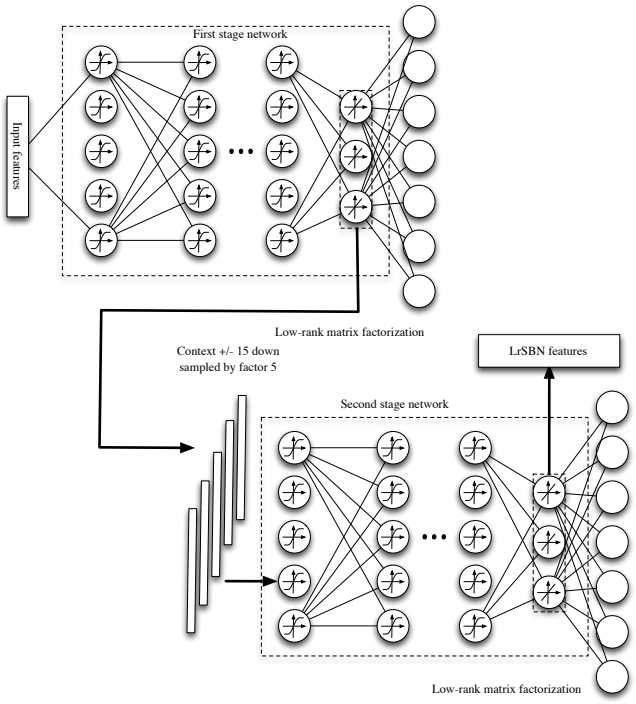

Figure 1: *Diagram of bottleneck feature extraction (from [7]).*

the second DNN are the outputs of the BN layer from the first DNN. Context expansion is achieved by concatenating frames with time offsets of -10, -5, 0, 5, and 10. Thus, the overall time context seen by the second DNN is 31 frames. Both DNNs use the same setup of 5 hidden sigmoid layers and 1 linear BN layer, and both use tied-states as target outputs. The targets are generated by forced alignment from an HMM baseline. No DNN pretraining is used. Finally, the raw BN outputs from the second DNN are whitened using a global PCA, and used as features for a conventional context dependent GMM-HMM system. More details about the architecture can be found in [7].

3. I-Vector Extraction

The I-vector approach was first introduced in speaker and language recognition [16], where a large GMM, called the universal background model (UBM), is typically trained to act as a prior model of the distribution of speech sounds. The speaker, or language-adapted UBM mean components, also known as a *supervector*, have been found to be an effective representation for speaker and language recognition. The I-vector approach models supervector adaptation to a given sequence of frames in a low dimensional space called the total variability space. In the I-vector framework, each speech utterance can be represented by a GMM supervector, which is assumed to be generated as follows:

$$M = m + Tw$$

where m is the speaker independent and channel independent supervector (which can be taken to be the UBM supervector), T is a rectangular matrix of low rank, and w is a random vector having a standard normal distribution prior $N(0, 1)$. The I-vector is a Maximum A Posteriori (MAP) point estimate of the latent variable w adapting the UBM (supervector m) to a given recording.

Recently, the I-vector method has been successfully applied to speaker and channel adaptation in speech recognition [12]. Adding speaker characteristics to the audio features allows the DNN to learn more efficiently how each speaker can produce a specific phoneme.

For this work, an I-vector extractor of dimension 100 has been trained, using a UBM consisting of 512 mixtures. The features used to train the UBM are 40 dimensional linear discriminant analysis (LDA)-transformed features of nine stacked MFCC frames of dimension 13. The fMLLR speaker adaptation transform has been applied to the feature vectors. In our experiments, the GMM sizes are much smaller than those used in [12] because the training data set is limited.

4. Experiments

4.1. Experimental Setup

The QCRI automatic Arabic speech recognition corpus consists of broadcast news reports and conversational shows spoken only in Modern Standard Arabic (MSA) [13]. The Al-Jazeera news channel is the main source for collecting these data which contains 60 hours of manually transcribed recordings. The recordings have been segmented and transcribed to avoid non-speech segments such as music and background noise. The recordings were made using satellite cable sampled at 16kHz. The development dataset consists of one hour of speech, and is composed of broadcast news reports. No conversational shows have been included. However, two hours of speech have been collected as a test set comprising both kinds of data types used in the training corpora.

In this work, MADA [17] is used to implement a morphological decomposition to normalize and vowelize the text by retrieving the missing diacritics. Since several vowelizations for a specific word are possible, a confidence score is provided for each candidate. MADA has been widely used for both Statistical Machine Translation [18, 19], and in ASR to address the aforementioned challenges. See [17] for more information regarding MADA operational details.

The lexicon has been created with phonological rules proposed by Biadsy *et al.* [20]. They describe rules for representing glottal stops, short vowels, coarticulation of the definite article Al, nunnation, diphthongs, word ending p (/t/), and case endings, while ignoring geminates.

The language model has been built from three different sources: 1) manual transcriptions of the training data, 2) automatically diacritized transcriptions (430K words) of Al-Jazeera broadcast news, and 3) automatically diacritized texts (109 million words) downloaded from the Al-Jazeera web site. The vocabulary contains 400K words, combining words from the audio transcriptions and the 400K most frequent words in Al-Jazeera web site texts. The development set has been used to choose the interpolation coefficient in the mixing of both text sources. The out-of-vocabulary (OOV) rate on the test set is 3.1%.

4.2. Basic GMM Systems

This section describes the basic ASR system used to produce the HMM state alignments that are needed for training the DNN from which BN features will be extracted. The first system is based on an HMM-GMM using conventional cepstral features. These acoustic features correspond to 12 Mel-frequency Cepstral Coefficients (MFCCs), energy, and their first and second derivatives. The trained speech recognizer contains 4,000 state

distributions, with a total of 128,000 Gaussian components.

The second system is also a GMM-based HMM system where speaker adaptation was applied to make the acoustic models more appropriate to a specific speaker. The method used in this work is fMLLR, a widely used technique for speaker adaptation proposed by Gales [10]. However, since the only speaker information available from the database is speaker turns, features are adapted on an utterance-by-utterance basis. This system uses a different feature set compared to the first ASR system. These features consist of stacking 13 MFCC speech frames with a context window of 9 frames. These features were then projected into a 40 dimensional space using an LDA transform.

	WER	
System	Dev. Set	Eval. Set
Basic GMM	28.01%	42.62%
GMM+fMLLR	19.04%	32.38%

Table 1: *WERs of GMM-Based systems*

The results obtained by the two systems are reported in Table 1. The frame labels used to train the DNN have been produced by the second ASR system.

4.3. Tandem BN-GMM ASR Systems

The architecture described in Section 2 has been used for the extraction of BN features. Several experiments have been conducted in order to measure the efficiency of speaker adaptation using I-vectors in the context of BN feature extraction. I-vectors can be appended to features at varying positions in this architecture: 1) appended to filter bank features, the input of the first DNN; 2) appended to concatenated BN layer outputs of the first DNN, which is the input of the second DNN, 3) finally, the I-vectors can be added to input features of both DNNs. Table 2 shows the word error rate (WER) results of the different experiments. Note that the development set on this table has not been used for training the DNNs. Instead, a part of the training set has been used to tune DNN parameters.

	WER	
System	Dev. Set	Eval. Set
GMM (no adaptation baseline)	15.84%	28.49%
GMM + fMLLR (adapt baseline)	14.04%	27.83%
GMM i-vec 1^{st} stage + fMLLR	15,05%	28.32%
GMM i-vec 2^{nd} stage	15.52%	28.81%
GMM i-vec 2^{nd} stage + fMLLR	**13.77%**	**26.65%**
GMM i-vec both stages + fMLLR	14.57%	28.18%

Table 2: *WERs with BN features in tandem BN-GMM systems.*

The input feature vectors for the tandem GMM systems are the concatenation of the MFCC and BN features on which the LDA procedure described above is applied. The first two lines of Table 2 show the WER for basic GMM ASR systems without using I-vectors. These form the no adaptation, and adaptation baselines for these experiments. Note that using BN features in addition to MFCC with GMM (second line to Table 2) led to an absolute WER improvement of 4.55% on the evaluation set over the basic MFCC GMM system using fMLLR adaptation presented in Table 1.

The last three lines show the results of using I-vectors in the configurations discussed before. The results show that using I-vectors in combination with filter bank features hurt the WER, regardless of whether or not the input of the second DNN was augmented by I-vectors. This result was somewhat surprising considering that the use of I-vectors in training of the first DNN led to a slightly better WER on the held out tuning set. On the other hand, appending I-vectors to the input features of the second DNN leads to a WER improvement of 1.18%, on the evaluation set, over the BNF baseline system (line 2 of Table 2). Note that the WER is worse than the baseline when no fMLLR is used with the I-vector adapted BN features (line 4 of the table). This could be the sign of a mismatch between the speaker adapted BN features and non-adapted MFCC-based features. Recall that the UBM used for extracting I-vectors has been trained with fMLLR adapted features.

4.4. Hybrid BN-DNN ASR Systems

The next experiments we conducted explored the use of BN features on hybrid DNN-based ASR systems. Two types of BN features were considered for these experiments. The first one, denoted as *bnf* is the usual set of BN features extracted from the second stage DNN, the same features used in the tandem BN-GMM experiments. The second option is to use the context expanded (over 30 frames, using one frame every 5) output of the first stage DNN in the LrSBN architecture as BN features as input of the DNN, as proposed in [6]. This features are referred as *cat* features.

The usual input for hybrid DNN training is the concatenation of 11 LDA frames consisting of a total of 440 features. This raises the question of whether the additional BN features have to be spliced or not. The first experiment conducted was to determine whether to splice or not. Note that in this experiment, MFCC-based features are spliced in the usual way.

	WER
System	Dev. Set
Spliced bnf	15.34%
Non-spliced bnf	**13.93%**
Spliced cat	15.52%
Non-Spliced cat	14.57%

Table 3: *Results with spliced and non spliced BN features.*

The experimental results, summarized in Table 3, show that a better WER is obtained when using BN features without any splicing, for both type of features. This result is not surprising considering that the DNN used to extract BN features consider a context of 31 frames. Consequently, contextual information are already present in BN features. Based on these results, splicing is applied only on MFCC-based features in all subsequent experiments. The input features will thus be the I-vector created from the utterance, appended to BN features and spliced LDA-fMLLR features. Figure 2 describes how features are combined in order to be fed into the DNN.

Table 4 shows the results of different experiments that have been conducted with different feature combinations. In this table, *bnf_ivec* denotes BN features trained with an I-vector appended to input features in the training process of the second DNN. The first experiment is a DNN fed with LDA-fMLLR features to which the I-vector extracted from the utterance has been appended. This is the baseline DNN experimental result

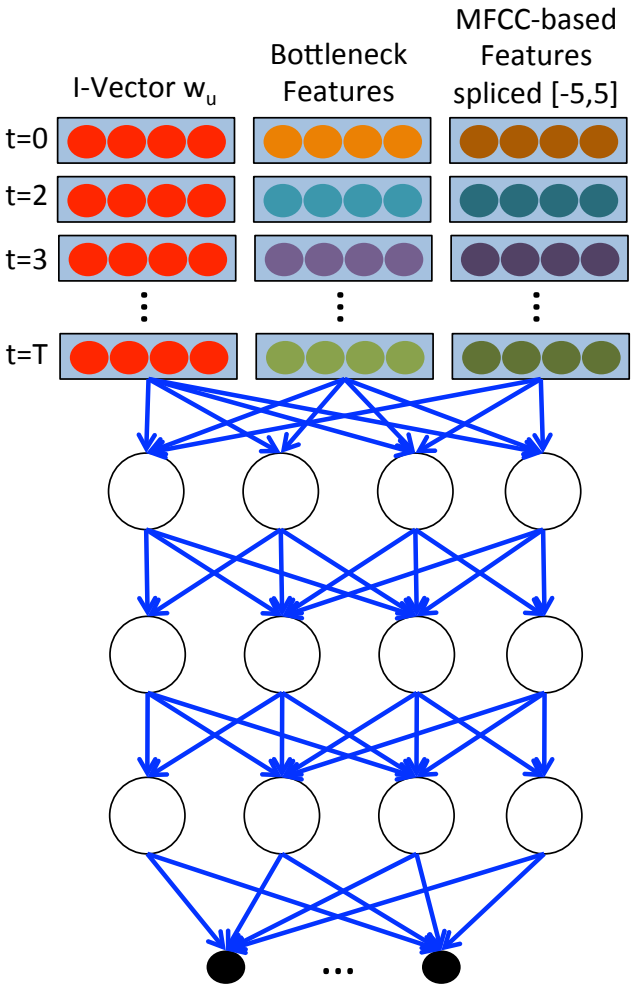

Figure 2: *Using I-vector as a speaker feature vector*

which shows that the fMLLR adapted tandem BN-GMM baseline system described in the previous section performs better than a normal DNN by 1.5% on the evaluation set. However, the use of sequential training with the MPE criterion works better, but with a difference of only 0.69% with BN features, and only 0.19% with *cat* features.

	WER	
System	Dev. Set	Eval. Set
fMLLR (adapted DNN baseline)	18.29%	29.49%
fMLLR + I-vector	17.33%	28.16%
fMLLR + I-vector+MPE	14.45%	25.96%
fMLLR + bnf	13.93%	26.41%
fMLLR + I-vector+bnf	14.91%	26.30%
fMLLR + bnf_ivec	14.66%	26.54%
fMLLR + I-vector+bnf_ivec	14.34%	26.24%
fMLLR + I-vector+bnf_ivec+MPE	14.00%	25.40%
fMLLR + cat	14.57%	26.08%
fMLLR + I-vector+cat	14,84%	25.89%
fMLLR + I-vector+cat+MPE	**13.63**%	**24.79**%

Table 4: *WERs with BN features in hybrid BN-DNN systems.*

The best result is obtained by using *cat* features with a sequentially trained DNN. In this case, the absolute WER on the evaluation set has been improved by 1.17% over the baseline (third line of Table 4). However, the effect of I-vectors on the overall WER is relatively small. The biggest part of the improvement comes from the use of BN features, as shown by the result with the DNN system without any use of I-vectors. Indeed, the results show that the improvement of using I-vector in both the BN feature extraction and the hybrid DNN setup is only 0.17% absolute on the evaluation set (line 4 and 7 in Table 4).

5. Conclusion

This paper presented a study on using I-vectors for speaker adaptation in the context of bottleneck features. The results show that adding I-vectors to the input features of the second level of the stacked bottleneck feature extraction architecture led to a WER improvement of 1.18% absolute on an Arabic broadcast and conversational speech task. The approach reduced the gap between tandem GMM and hybrid DNN systems to only 0.69% absolute.

Experimental results also confirm previous work that using BN features in a DNN-based system improves the WER. The improvement is up to 1.16% absolute in the scenario presented in this work. The use of I-vectors in a hybrid DNN using BN features has a small effect on the WER, compared to results published so far. This may be due to the fact that BN features are able to capture similar information about speakers. We plan to investigate this further in future work.

6. Acknowledgments

This research was supported by the Qatar Computing Research Institute (QCRI). We would like to thank Tuka Al Hanai for her help with pronunciation modeling, and Ahmed Ali for his collaboration on this project.

7. References

[1] H. Hermansky, D. Ellis, and S.Sharma, "Tandem connectionist extraction for conventional hmm systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2000, pp. 1635–1639.

[2] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2011, p. 273240.

[3] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, p. 41534156.

[4] Z. J. Yan, Q. Huo, and J. Xu, "A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2013.

[5] M. Karafiat, F. Grezl, M. Hannemann, K. Vesely, and J. H. Cernocky, "BUT Babel system for spontaneous Cantonese," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2013.

[6] M. Karafiat, F. Grezl, K. Vesely, M. Hannemann, I. Szoke, and J. H. Cernocky, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2014.

[7] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factor-

ization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

[8] J. Gehring, W. Lee, K. Kilgour, L. I, Y. Miao, and A. Waibel, "Modular combination of deep neural networks for acoustic modeling," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2013.

[9] S. P. Rath, K. Knill, A. Ragni, and M. J. F. Gales, "Combining tandem and hybrid systems for improved speech recognition and keyword spotting on low resource languages," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2014.

[10] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," in *Computer Speech and Language*, vol. 12, 1998, pp. 75–98.

[11] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP 2013*, 2013.

[12] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 55–59.

[13] P. Cardinal, A. Ali, N. Dehak, Y. Zhang, T. A. Hanai, Y. Zhang, J. Glass, and S. Vogel, "Recent advances in ASR applied to an Arabic transcription system for Al-Jazeera," in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2014.

[14] F. Grzl and M. Karafit, "Hierarchical neural net architectures for feature extraction in ASR," in *Proceedings of 10th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2010, pp. 1201–1204.

[15] T. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6655–6659.

[16] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[17] N. Habash, O. Rambow, and R. Roth, "Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, 2009, pp. 102–109.

[18] A. El Kholy, N. Habash, G. Leusch, E. Matusov, and H. Sawaf, "Language independent connectivity strength features for phrase pivot statistical machine translation," *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 13, 2013.

[19] M. Carpuat and M. Diab, "Task-based evaluation of multiword expressions: a pilot study in statistical machine translation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2010, pp. 242–245.

[20] B. Fadi, N. Habash, and J. Hirschberg, "Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics*, 2009.