END-TO-END NEURAL NETWORK BASED AUTOMATED SPEECH SCORING

Lei Chen, Jidong Tao‡, Shabnam Ghaffarzadegan, Yao Qian†*

Liulishuo Silicon Valley AI Lab, 1900 S Norfork St, San Mateo, CA 94403, USA
‡ Midea America Corporation, 250 W Tasman Dr, San Jose, CA 95134, USA
* Robert Bosch Corporation, 4005 Miranda Ave, Palo Alto, CA 94304, USA
† Educational Testing Service (ETS), 90 New Montgomery St, San Francisco, CA, 94105, USA
lei.chen@liulishuo.com, vjdtao@hotmail.com, shabnam.ghaffarzadegan@us.bosch.com, yqian@ets.org

ABSTRACT

In recent years, machine learning models for automated speech scoring systems were mainly built using data-driven approaches with handcrafted features as one of the main components. However, the remarkable successes of deep learning (DL) technology in a variety of machine learning tasks has demonstrated its effectiveness in extracting features. Although there have been some efforts in utilizing DL technology for the automated speech scoring task, a thorough investigation of learning useful features is still missing. In this paper, we propose an end-to-end solution that consists of using deep neural network models to encode both lexical and acoustical cues to learn predictive features automatically. Experiments also confirm the effectiveness of our proposed solution compared to conventional methods based on handcrafted features.

Index Terms— automatic speech scoring, deep neural network, CNN, LSTM, attention.

1. INTRODUCTION

[1]In the last two decades, there have been a large number of studies using automatic speech recognition (ASR) technology to support language learning, such as computer aided pronunciation training (CAPT) and automated speech scoring (see [1] for a comprehensive review). In an automated speech scoring system, as exemplified in [2, 3], different handcrafted speech features were computed using various methods including signal processing, prosodic analysis, and natural language processing (NLP). The extracted features were fed into a statistical model to predict the scores reflecting speaking proficiency levels.

However, handcrafted features are not an ideal choice due to the difficulties in finding the right features for the task and the substantial development effort. Recently, many machine learning tasks deploy end-to-end methods, which automatically learn features, and use a coherent process jointly obtain representations and models. These end-to-end solutions have shown advantages in achieving a more efficient model-building process and improved prediction performance. Clearly, such solutions suggest a promising direction for the automated speech scoring field as well.

This paper is organized as follows: section 2 briefly reviews previous research using Deep Learning (DL) based ASR systems and scoring methods for automated scoring tasks; section 3 describes our proposed end-to-end solution using different neural network (NN) models; section 4 compares our end-to-end solution with the conventional method of using handcrafted features; finally, section 5 makes conclusions and suggests future research directions.

[1]This work was done while the first and second authors worked as full time employees at ETS. The third author participated in this work's early stage research as a summer intern at ETS while she was affiliated with University of Texas at Dallas.

2. PREVIOUS RESEARCH

Speech scoring is the task of measuring speech proficiency based on a set of predefined features suggested in English Language Learner (ELL) studies, including speaking fluency, intonation, vocabulary, etc. Most of the previous work on measuring speech proficiency used ASR outputs and prosodic analyses to calculate the score. SpeechRater^{SM} for the Educational Testing Service® (ETS) TOEFL® Practice Test Online (TPO) is a working example of this method using a rich set of handcrafted speech features [2].

In recent years, fast growing DL technology has also been applied to the speech scoring task. Beyond providing more accurate recognition outputs, acoustic models (AMs) using deep neural network (DNN) structures have been largely used to improve pronunciation measurements [4–8]. For example, [4, 5] used a deep belief network (DBN) model as AMs and found that DBN AMs improved pronunciation evaluation performance over their GMM counterparts. [6, 7] investigated the use of context-dependent DNN hidden Markov models (CD-DNN-HMM), to improve ASR, and obtained more accurate automatic assessment of child English learners. [8] investigated using three types of DL based AM structures, i.e., DNN, Convolution Neural Network (CNN) [9], and a Tandem GMM-HMM model using bottleneck features. These DL-based AMs were found to provide substantial increases in recognition accuracy and improved scoring performance compared to GMM AMs.

Moreover, there have been several successful applications of deep learning based automated scoring models to written responses. [10] proposed an end-to-end NN-based model to automatically score essays. The model contained a special word embedding training part that considered the essays' scores to be additional constrains and a bi-directional Recurrent Neural Network (RNN) for learning features [11]. On the Automated Student Assessment Prize (ASAP) essay data set, this NN scoring model showed better performance than the conventional model that used handcrafted features, e.g., word and part-of-speech (POS) n-grams, phrase-structure, etc. On the same ASAP essay data set, [12] proposed a hybrid NN model that consisted of a CNN model to encode local context information and an RNN to encode long-range information. Instead of using the RNN model's hidden vector on the last time stamp, a mean over time (MOT) aggregation was used to utilize information over the en-

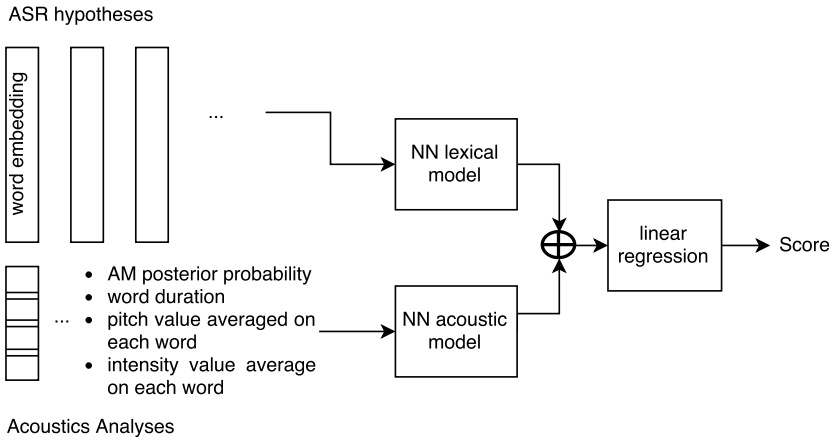

Fig. 1. A diagram showing our proposed end-to-end automated scoring solution using two NN encoders for feature learning, one for lexical information and the other for acoustic information

tire essay. The proposed NN model showed improved performance compared to a publicly available open-source conventional model, Enhanced AI Scorer (EASE)[2]. On spoken responses, a preliminary trial of the end-to-end scoring was made in [13]. Using prosodic and spectrum measurements, a bi-directional RNN was used to learn features. However, the learned features were tested together with the handcrafted features in [13], and it was not clear whether the learned features could work independently or not. In this paper, we investigate end-to-end scoring solution for the automated speech scoring task using both lexical cues in recognized words and acoustical cues to rate open-ended spoken responses.

3. DEEP LEARNING BASED SCORING MODELS

Figure 1 depicts our end-to-end automated speech scoring solution. Two DL-based models are used to encode both lexical and acoustic cues. The encoded features are concatenated and fed into a linear regression model to predict the scores. To build the lexical model, the word tokens being recognized are converted into input tensors using a word embedding layer. For acoustic model, we use four measurements for each word: (1) AM posterior probability, (2) word duration, (3) mean value of pitch, and (4) mean value of intensity. These measurement are chosen because they are the widely used cues, which will be explained in more details in Section 4. Moreover, three different deep learning structures are used to model the learned word representation for lexical and acoustical cues, including: 1D CNN, Bi-Directional RNN using Long Short-Time Memory (LSTM) [11] cells (BD-LSTM), and the BD-LSTM RNN with an attention weighting scheme.

3.1. CNN based scoring model

Subplot (a) in Figure 2 shows the details of the CNN model used in this study. After receiving inputs, a dropout layer with probability $dp_{CNN}1$ is applied before a $1D$ convolution. Following [14], convolution filters with varied sizes ($conv_{size} - 1$, $conv_{size}$, and $conv_{size} + 1$) are used to cover different receiving fields. For each size, $conv_n$ filters are used. For each filter output, a max-over-time pooling layer is used, which results in a $3 \times conv_n$ dimensional encoded vector. This vector runs through a second dropout layer with probability ($dp_{CNN}2$). Finally, the entire output of the CNN encoder is fed into a linear regression model to predict speech score.

3.2. BD-LSTM based scoring model

An RNN model processes a sequence of input data by recursively applying a transitional function to its hidden state vector $\mathbf{h}_t$. The activation of $\mathbf{h}_t$ at time-step t depends on both the current input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$.

$$\mathbf{h_t} = \mathbf{f}(\mathbf{h_{t-1}}, \mathbf{x_t}) \tag{1}$$

Commonly, an RNN model encodes an input sequence to a fixed-sized vector $\mathbf{h}_T$ on its last time step T and uses it as the input for following prediction steps. Using an RNN alone can be hampered by the *exploding or vanishing gradients* problem, which is the fact that gradients may grow or decay exponentially during RNN training. An LSTM cell addresses this issue, and makes RNNs useful in practice [11]. As a result an LSTM structure is also used in our study. We describe the implementation following [15].

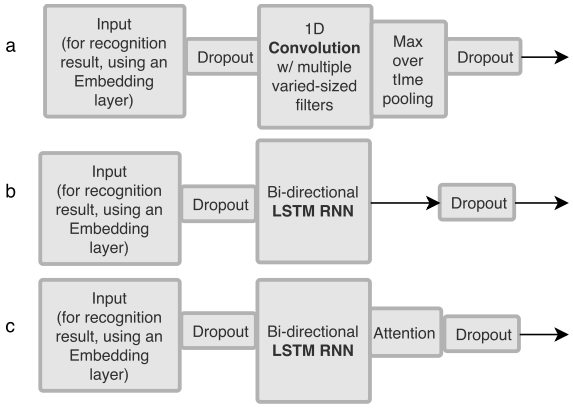

Fig. 2. Three types of NN models for encoding input cues to learn useful features.

[2] https://github.com/edx/ease

LSTM units at each time step t can be defined by a collection of vectors in $\mathbb{R}^d$: an *input gate* $\mathbf{i}_t$, a *forget gate* $\mathbf{f}_t$, an *output gate* $\mathbf{o}_t$, a *memory cell* $\mathbf{c}_t$ and a hidden state $\mathbf{h}_t$. At each time step, an LSTM maintains a hidden vector $\mathbf{h}$ and a memory vector $\mathbf{c}$, responsible for controlling state updates and outputs. More concretely, we define the computation at time step t as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1}) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{c}_{t-1}) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_t) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1}) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c_t})
\end{aligned} \tag{2}$$

where $\mathbf{x}_t$ is the input (word embedding vector) at the current time step t, σ is the logistic sigmoid function and $\odot$ denotes element-wise multiplication. The forget gate $\mathbf{f}_t$ controls how much of the information stored in the memory cell will be erased, the input gate $\mathbf{i}_t$ controls how much each unit is updated, and the output gate $\mathbf{o}_t$ controls the exposure of the internal memory cell.

Part (b) of Figure 2 shows the BD RNN with LSTM cells used in our study. A bi-directional network is chosen here to take into account information both from past and future given the inherent nature of speech and language production. The final vector of each hidden state at time t is formed by concatenating the hidden state vectors from two directions with $LSTM_{dim}^{cue}$ dimensions. In this context, cue can refer to either lexical (lex) or acoustical (ac) information. Note that two dropout layers are applied before and after the BD RNN layer with the probability values of $dp_{RNN}1$ and $dp_{RNN}2$.

3.3. BD-LSTM attention scoring model

In this section, we introduce the attention mechanism of our BD-LSTM model which is proven very effective in many natural language processing tasks [16]. As Figure 2 (c) shows, an attention model is added to our system through one more layer. In the BD-LSTM model, only the last hidden state ($\mathbf{h}_T$) is used to make the final decision, and the context information from previous times (prior to T) were not utilized. To overcome this limitation, we use a simple feed-forward attention model as proposed in [17] to obtain a set of weights for all hidden states. A single vector $\mathbf{S}$ from the entire sequence ($\mathbf{h}_t$) can be formulated as follows:

$$\mathbf{e_t} = \mathbf{a}(\mathbf{h_t}), \alpha_\mathbf{t} = \frac{\mathbf{exp}(\mathbf{e_t})}{\sum_{\mathbf{k=1}}^{\mathbf{T}} \mathbf{exp}(\mathbf{e_k})}, \mathbf{S} = \sum_{\mathbf{t=1}}^{\mathbf{T}} \alpha_\mathbf{t}\mathbf{h_t} \tag{3}$$

where, a is a learnable function depending on $\mathbf{h}_t$. This simplified feed-forward attention can be seen as producing a fixed-length embedding $\mathbf{S}$ of the input sequence by computing an adaptive weighted average of the hidden state sequence $\mathbf{h}$. Figure 3 represents more details on the attention mechanism.

4. EXPERIMENTS

4.1. Database

In this study, we utilize the data collected from an online practice for a well known English test that measures test takers' readiness to attend schools with English as the primary instructional language. The dataset is divided into three partitions: the *train* set containing 2930 spoken responses, the *dev* set containing 731 responses, and the *eval* set containing 1827 responses. All spoken responses were scored by experienced human raters following a 4-point scale scoring rubric designed for scoring this English test. A score of 1 was the lowest band while a score of 4 was the highest band. Each response was scored by one group of raters (R1) and verified by a second group of raters (R2). Note that in our experiments, only the scores provided by the R1 group were used as ground truth scores.

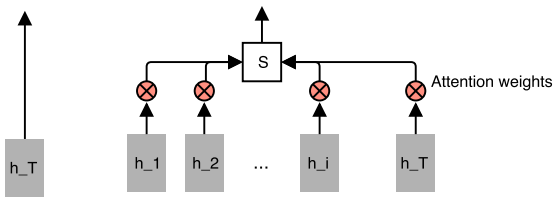

Fig. 3. Using BD-LSTM's last hidden state vs. using a feed forward attention mechanism to utilize a weighted average of all hidden states.

The ASR system used for recognizing test takers' non-native English responses is a DNN-HMM hybrid ASR system built with the Kaldi open-source toolkit. This model is a 5-layer feed-forward DNN using acoustic features from the current frame plus the previous and following 5 context frames. More details can be found in [8]. The ASR model is trained on transcribed responses from the operational English test containing 819 hours of non-native spontaneous speech covering more than 100 first languages from about 150 countries around the world. As reported in [8], this ASR system achieved a word error rate of 35% on the spoken responses collected in the online practice test.

4.2. Conventional model

SpeechRater^{SM}, an automated scoring engine for assessing non-native English proficiency [2], was used to extract scoring features. The features are related to several aspects of the speaking construct[3], which include *fluency*, *rhythm*, *intonation* & *stress*, *pronunciation*, *grammar*, and *vocabulary* use. Table 1 provides a concise synopsis of these features.

Using Pearson correlations between these features and human rated scores computed on the train set, a subset of features ($n = 12$) were selected. Then, the SKLL toolkit[4] was used for training and evaluating different prompt-independent scoring models. We run different regression methods, including Random Forest (RF), Gradient Boosting Tree (GBT), Support Vector Regression (SVR) for the speech scoring task. The hyper-parameters of these models were decided automatically by the SKLL using a 5-fold cross validation on the train set. Among the three methods, the GBT model was found to provide the highest machine-human score correlation.

4.3. Deep learning based models

We used the Keras Python package to implement all DL-based models described in Section 3. We used pre-trained GloVe word embeddings [21] and set the embedding dimension to 300. When a word could not be found in the GloVe embeddings' vocabulary, we set up its embedding vector to be all zeros. The embedding vectors were further fine-tuned during model training steps. For acoustic cues, we used Kaldi ASR's outputs to obtain both AM posterior probabilities

[3] In psychometric terms, a *construct* is a set of knowledge, skills, and abilities that are required in a given domain.

[4] `https://github.com/EducationalTestingService/skll`

Category	Example Features
Fluency	Features based on the number of words per second, number of words per chunk, number of silences, average duration of silences, frequency of long pauses ($\geq$ 0.5 sec.), number of filled pauses (*uh* and *um*) [2], frequency of between-clause silences and edit disfluencies compared to within-clause silences and edit disfluencies [18].
Rhythm, Intonation & Stress	Features based on the distribution of prosodic events (prominences and boundary tones) in an utterance as detected by a statistical classifier (overall percentages of prosodic events, mean distance between events, mean deviation of distance between events) [2] as well as features based on the distribution of vowel, consonant, and syllable durations (overall percentages, standard deviation, and Pairwise Variability Index) [19].
Pronunciation	Acoustic model likelihood scores, generated during forced alignment with a native speaker acoustic model, the average word-level confidence score of ASR and the average difference between the vowel durations in the utterance and vowel-specific means based on a corpus of native speech [20]
Grammar	Similarity scores of the grammar of ASR output of the response with respect to reference response.
Vocabulary Use	Features about the diversity and sophistication of the vocabulary based on the ASR output.

Table 1. Descriptions of SpeechRater^{SM} features for automated speech scoring.

and durations, and we used Praat[5] software to obtain pitch and intensity measurements. When training the DL-based models using the Adam optimization [22], we randomly selected 10% of the train set for early stopping to avoid over-fitting. For DL hyperparameter tuning, Tree Parzen Estimation (TPE) method [23] was utilized. This approach was implemented using the Hyperopt Python package. We run Keras with the Theano backend on an Nvidia Titan X GPU card to speed up the entire experiment. After running 100 iterations of hyperparameter search, we ended up with the following selection: $conv_{size}$ is 4 (which entails that the various filter sizes were $(3, 4, 5)$), $conv_n$ is 100, $dp_{cnn}1$ is 0.25, $dp_{cnn}2$ is 0.5, $LSTM_{dim}^{lex}$ is 128, $LSTM_{dim}^{ac}$ is 32, $dp_{LSTM}1$ is 0.25, and $dp_{LSTM}2$ is 0.5.

4.4. Results

Table 2 reports our machine scoring experiment using both the conventional method and the DL-based methods explained in section 4.2 and 4.3. The conventional model using sophisticated speech features and the GBT regression model leads to a Pearson correlation of 0.585 between the machine-predicted scores and the human-rated scores. This result was consistent with previously reported results on similar tasks, such as [8]. Our CNN based model achieved a Pearson correlation of 0.581, which is very close to the conventional model. Moreover, the BD-LSTM model did not show any considerable performance improvement, in spite of incorporating richer sequential information, as compared to the CNN model. However, after applying the simple feed-forward attention mechanism, the predicted scores had the highest Pearson correlation with the human rated ones, at $r = 0.602$ correlation. This result shows that weighting among all internal hidden states plays an important role in increasing prediction accuracy for the speech scoring task. In other words, the machine needs to focus on specific part of the speech to evaluate the proficiency, instead of taking into account the whole response. This fact applies to human raters as well. Our experimental results confirm the power of DL in extracting meaningful representations for the speech scoring task, which has superior performance compared to the handcrafted features, which were developed during a long time of research in both the second language learning and automated assessment fields. Note that the information resources provided to these two models are not equal. For example, word embedding representations, which are viewed as providing better lexical presentations than n-grams, were used in the NN model. Therefore, the performance gain may be caused by a compound factor of using both rich information and the neural network structure.

System	Pearson r
Conventional model	0.585
CNN	0.581
BD-LSTM	0.531
BD-LSTM w/ attention	0.602

Table 2. A comparison of the Pearson correlations between human rated scores and the machine predicted scores from the conventional model and the NN models using different encoders

[5] http://www.fon.hum.uva.nl/praat/

5. CONCLUSIONS

In this study, we investigated deep learning based technology to solve the automated speech scoring task in an end-to-end approach. To our knowledge, this is the first study to use automatically induced features from both ASR hypotheses and basic acoustic analyses. We studied different DL models to learn the best predictive features for the speech scoring task. In particular, the CNN model showed a scoring performance quite close to the one demonstrated by a conventional method using handcrafted features and a GBT regression model. When using an attention mechanism to utilize all the hidden states' information, the BD-LSTM model showed a dramatic performance improvement compared to both traditional and other DL-based models. Experimental results confirm the effectiveness of end-to-end solutions for automated assessment research.

This study leads to the following steps for future improvement. Firstly, it will be important to increase the explainability of DL-based models. Next, more acoustic cues can be utilized to provide a comprehensive coverage. Finally, other sophisticated attention mechanisms can be explored to improve the performance.

6. REFERENCES

[1] M. Eskenazi, "An overview of spoken language technology for education," *Speech Communication*, vol. 51, no. 10, pp. 832–844, 2009.

[2] K. Zechner, D. Higgins, X. Xi, and D. M. Williamson, "Automatic scoring of non-native spontaneous speech in tests of spoken english," *Speech Communication*, vol. 51, pp. 883–895, October 2009.

[3] J. Bernstein, A. Van Moere, and J. Cheng, "Validating automated speaking tests," *Language Testing*, vol. 27, no. 3, pp. 355, 2010.

[4] W. Hu, Y. Qian, and F. K. Soong, "A new DNN-based high quality pronunciation evaluation for computer-aided language learning (CALL).," in *INTERSPEECH*, pp. 1886–1890.

[5] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166.

[6] A. Metallinou and J. Cheng, "Using deep neural networks to improve proficiency assessment for children english language learners," in *Proc. of INTERSPEECH*, pp. 1468–1472.

[7] J. Cheng, X. Chen, and A. Metallinou, "Deep neural network acoustic models for spoken assessment applications," *Speech Communication*, vol. 73, pp. 14–27.

[8] J Tao, S Ghaffarzadegan, and L Chen, "Exploring deep learning architectures for automatically grading non-native spontaneous speech," in *ICASSP*, Shanghai, China, 2016.

[9] Y LeCun and K Kavukcuoglu, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 2010.

[10] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei, "Automatic Text Scoring Using Neural Networks," in *Proc. of ACL*, jun 2016.

[11] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[12] Kaveh Taghipour and Hwee Tou Ng, "A neural approach to automated essay scoring.," in *EMNLP*, 2016, pp. 1882–1891.

[13] Zhou Yu, Vikram Ramanarayanan, David Suendermann-Oeft, Xinhao Wang, Klaus Zechner, Lei Chen, Jidong Tao, Aliaksei Ivanou, and Yao Qian, "Using bidirectional lstm recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. dec 2015, pp. 338–345, IEEE.

[14] Y Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1746–1751.

[15] A Graves, S Fernández, and J Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," *Artificial Neural Networks:*, 2005.

[16] Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[17] Colin Raffel and Daniel P. W. Ellis, "Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems," *arXiv preprint arXiv:1512.08756*, Dec. 2015.

[18] Lei Chen and Su-Youn Yoon, "Application of structural events detected on asr outputs for automated speaking assessment.," in *INTERSPEECH*, 2012.

[19] Lei Chen and Klaus Zechner, "Applying rhythm features to automatically assess non-native speech.," in *INTERSPEECH*, 2011, pp. 1861–1864.

[20] L. Chen, K. Zechner, and X Xi, "Improved pronunciation features for construct-driven assessment of non-native spontaneous speech," in *NAACL-HLT*, 2009.

[21] J Pennington, R Socher, and CD Manning, "GloVe: Global Vectors for Word Representation.," in *EMNLP*, 2014.

[22] D Kingma and J Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] James Bergstra, Daniel Yamins, and David D Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures.," *ICML (1)*, vol. 28, pp. 115–123, 2013.