USING BIDIRECTIONAL LSTM RECURRENT NEURAL NETWORKS TO LEARN HIGH-LEVEL ABSTRACTIONS OF SEQUENTIAL FEATURES FOR AUTOMATED SCORING OF NON-NATIVE SPONTANEOUS SPEECH

Zhou Yu, Vikram Ramanarayanan*[†]*, David Suendermann-Oeft*[†]*, Xinhao Wang*[†]*,*
Klaus Zechner[‡]*, Lei Chen*[‡]*, Jidong Tao*[‡] *and Yao Qian*[†]

Educational Testing Service R&D
[†] 90 New Montgomery St, 1500, San Francisco, CA
[†] 660 Rosedale Road, Princeton, NJ
[*] Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA

ABSTRACT

We present a technique to jointly learn the high level abstractions of sequential features (such as, pitch and MFCC's) and combine time-aggregated features (such as, mean length of pauses, recognizer confidence scores, etc.) to optimize the automated scoring of non-native spoken responses. We use a bidirectional long short term memory (BLSTM) network, a type of recurrent neural network to optimize the scoring process jointly by learning the high-level abstractions of the time-sequence features together with traditional time-aggregated features. We find such model reaches the best performance in terms of correlation with human raters. We also find incorporating time-sequence features improves the performance drastically when there are limited time-aggregated features. Thus reducing the effort and resource in generating these fine-grained features for automated scoring.

Index Terms— Automatic speech scoring, non-native speech, recurrent neural networks

1. INTRODUCTION AND RELATED WORK

Receptive language skills, i.e., reading and listening, are typically assessed using a multiple-choice paradigm, while productive skills , i.e., writing and speaking, usually are assessed by eliciting constructed responses from the test taker. Constructed responses are written or spoken samples such as essays or spoken utterances in response to certain prompt and stimulus materials in a language test. Due to the complexity of the constructed responses, scoring has been traditionally performed by trained human raters, who follow a rubric that describes the characteristics of responses for each score point. However, there are a number of disadvantages associated with human scoring, including factors of time and cost, scheduling issues for large-scale assessments, rater consistency, rater bias, central tendency, etc. [1]. To overcome these limitations, systems for automated scoring of constructed responses have been developed, both for the written and the spoken domain of language proficiency assessments [2, 3, 4, 5, 6, 7].

Most automated scoring systems involve two main stages: (1) computation of manually engineered features describing certain aspects of the language proficiency that is to be assessed, using natural language processing and related technologies; and (2) a scoring model which combines these features using supervised machine learning with human scores as criterion variables to generate a score for each constructed response.

Many state-of-the art automated speech scoring systems leverage an automatic speech recognition (ASR) front-end system that provides word hypotheses about what the test taker said in his/her response. As one might be expected, training such a system requires a large corpus of non-native speech as well as manual transcriptions thereof. The outputs of this front-end are then used to design further features (lexical, prosodic, semantic, etc.) specifically for automatic speech assessment, which are then fed into a machine-learning-based scoring model. Previous work has typically optimized these two stages independently. For example, response-level features specifically suited to the scoring task are manually engineered, and then fed into simple machine learning models, such as linear regression [8] or CART trees [9] to obtain the final score. However, to our knowledge, no work so far has investigated learning the features and optimizing the scoring model *jointly*.

With the advent of high-performance deep neural networks in recent years, it has become possible to utilize their power to automatically abstract from a low-level feature representation to generate higher-level features, i.e., without human expert knowledge being involved, that could provide additional information to a scoring model that is originally

The first author performed the work while she was interning with Educational Testing Service R&D in San Francisco, CA.

built solely on expert-engineered features. Automated feature induction using deep neural network approaches have been used successfully already in other domains, such as, object recognition [10], or multimodal analysis [11].

We propose to use Bidirectional Long Short Term Memory Recurrent Neural Networks (BLSTM) to combine different features for scoring spoken constructed responses. BLSTMs allow us to to capture information regarding the spatiotemporal structure of the input spoken response time-series. In addition, by using a bidirectional optimization process, both past and future context are integrated into the model. Finally, by combining higher-level abstractions obtained from the BLSTM model with time-aggregated response-level features, we aim to design an automated scoring system that utilizes both time-sequence and time-aggregated information from speech to achieve optimal performance.

The rest of the paper is structured as following: we first introduce our data set and features in Sections 2 and 3 respectively, and then describe the BLSTM model (Section 4) and the specific network architecture adapted for the automated spoken response scoring task in Section 5. Finally we report our results of our experimental analysis in Section 6.

2. DATA

The data used in this study is drawn from an international assessment for non-native speakers, which measures the ability to use and understand English at the university level. Its speaking section consists of two different types of test questions to elicit spontaneous speech, referred to as independent and integrated tasks. The independent tasks require test takers to express their opinions on a familiar topic in the form of a 45-second spoken response, while the integrated tasks require them to speak a 60-second spoken response based on reading and listening to relevant prompt materials.

Human experts were recruited to rate the overall proficiency scores on a 4-point scale, which addresses three main aspects of speaking proficiency, including delivery, language use, and topic development. For example, based on a general description of the human scoring rubrics, a score of 4 (highest level) indicates that the response fulfills the demands of the task, with at most minor lapses in completeness. It is highly intelligible and exhibits sustained and coherent discourse; a score of 1 (lowest level), on the other hand, indicates that the response is very limited in content and/or coherence, or is only minimally connected to the task, or the speech is largely unintelligible. The whole dataset is randomly split into three partitions: 12,593 responses for training, 2,000 responses for development, and 1,363 responses for evaluation.

3. FEATURE DESCRIPTIONS

We combine fine-grained, time-aggregated features at the level of the entire response that capture pronunciation, grammar, etc. (that the SpeechRater system [3] produces) with time-sequence features that capture frame-by-frame information regarding prosody, phone content and speaker voice quality of the input speech. We use a BLSTM with either a multilayer perceptron (MLP) or a linear regression (LR) based output layer to jointly optimize the automated scoring model.

3.1. Time-Aggregated Feature Descriptions

SpeechRater extracts a range of features related to several aspects of the speaking construct.[2] These include pronunciation, fluency, intonation, rhythm, vocabulary use, and grammar. A selection of 91 of these features are used to score spontaneous speech and all of them are generic as opposed to being designed specifically for certain test questions. See Table 1 for a concise synopsis of these features. We refer to this set of 91 features as the **Content** feature set. Within the **Content** feature set, there is a subset of features that only consist of meta information, such as the length of the audio file, the gender of the test taker, etc. We refer to this set of 7 features as the **Meta** feature set.

3.2. Time-Sequence Feature Descriptions

The time-aggregated features computed from the input spoken response take into account delivery, prosody, lexical and grammatical information. Among these, features such as the number of silences capture aggregated information over time. However, previous work has found that some pauses might be more salient than others for purposes of scoring – for instance, silent pauses that occur at clause boundaries in particular are highly correlated with language proficiency grading [16]. In addition, time-aggregated features do not fully consider the evolution of the response over time. Thus we introduce *time-sequence* features that contain attempt to capture the evolution of information over time and use machine learning methods to discover structure patterns in this information stream. We extract six prosodic features – "**Loudness**", "**F0**", "**Voicing**", "**Jitter Local**", "**Jitter DDP**" and "**Shimmer Local**". "**Loudness**" captures the loudness of speech, i.e., the normalised intensity. "**F0**" is the smoothed fundamental frequency contour."**Voicing**" stands for the voicing probability of the final fundamental frequency candidate, which captures the breathy level of the speech. "**Jitter Local**" and "**Jitter DDP**" are measures of the frame-to-frame jitter, which is defined as the deviation in pitch period length, and the differential frame-to-frame jitter, respectively."**Shimmer Local**" is

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

Category	Sub-category	Quantity	Example Features
Prosody	Fluency	19	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 (≥ 0.5 sec.), number of filled pauses (*uh* and *um*) [3].
	Pitch & Power	11	Basic descriptive statistics (mean, minimum, maximum, range, standard deviation) for the pitch and power measurements for the utterance.
	Rhythm, Intonation & Stress	12	Features based on the distribution of prosodic events (promincences 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) [3] as well as features based on the distribution of vowel, consonant, and syllable durations (overall percentages, standard deviation, and Pairwise Variability Index) [12].
Pronunciation	-	11	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 [13]
Disfluencies	–	6	Frequency of between-clause silences and edit disfluencies compared to within-clause silences and edit disfluencies [14, 15].
Grammar	–	12	Similarity scores of the grammar of the response in ASR with respect to reference response.
Vocabulary Use	–	13	Features about how diverse and sophisticated the vocabulary based on the ASR output.
Item Meta Info	–	7	The length of response in seconds, test taker's gender, test location, native country and native language. Response type, which is independent or dependent, and the index of the response.

Table 1. Descriptions of time-aggregated features

the frame-to-frame shimmer, which is defined as the amplitude deviation between pitch periods.

Apart from prosodic features, we also extracted a group of "**Mel-Frequency Cepstrum Coefficients**" (MFCC's) from 26 filter-bank channels. MFCC's capture an overall timbre parameter which measures both what is said (phones) and the specifics of the speaker voice quality, which provides more speech information apart from the prosodic features we mentioned above. We computed MFCCs using a frame size of 25ms and a frame shift size of 10ms, based on the configuration files provided in [17]. We use the first 13 of the coefficients for out experiments. MFCC features are popularly used in phoneme classification, speech recognition or higher level multimodal social signal processing tasks [18].

4. BLSTM MODEL DESCRIPTION

Long Short Term Memory Recurrent Neural Networks (LSTM) have been proved to be a successful attempt [19] to address the problem of the vanishing gradients for Recurrent Neural Networks (RNNs). The LSTM architecture consists of a set of recurrently connected subnets, known as memory blocks. Each block contains one or more self-connected memory cells and three multiplicative units - the input, output and forget gates - that provide continuous analogues of write, read and reset operations for the cells. A LSTM network is formed exactly like a simple RNN, except that the nonlinear units in the hidden layers are replaced by memory blocks.

The multiplicative gates allow LSTM memory cells to store and access information over long periods of time, thereby avoiding the vanishing gradient problem. For example, as long as the input gate remains closed (i.e. has an activation close to 0), the activation of the cell will not be overwritten by the new inputs arriving in the network, and can therefore be made available to the net much later in the sequence, by opening the output gate.

Given an input sequence x = $(x_1, ..., x_T)$, a standard recurrent neural network (RNN) computes the hidden vector sequence h = $(h_1, ..., h_T)$ and output vector sequence y = $(y_1, ..., y_T)$ by iterating the following equations from t = 1 to T:

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

where the W terms denote weight matrices (e.g. W_{xh} is the input-hidden weight matrix), the b terms denote bias vectors (e.g. b_h is the hidden bias vector) and $\mathcal{H}$ is the hidden layer function. $\mathcal{H}$ is usually an element wise application of a sigmoid function. However we have found that the Long Short-

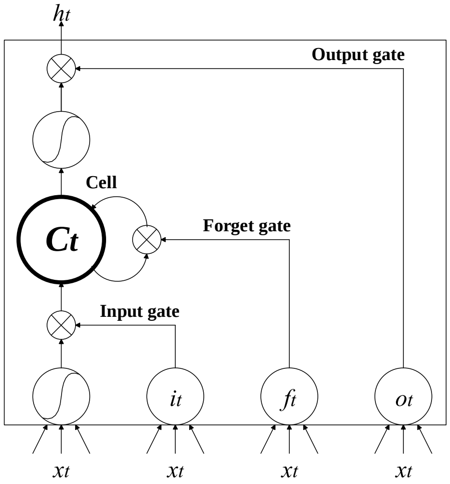

Fig. 1. Long Short-term Memory Cell.

Term Memory (LSTM) architecture [19], which uses custom-built memory cells to store information, is better at finding and exploiting long range context. Figure 1 shows a single LSTM memory cell.
For the version of LSTM used in this paper, $\mathcal{H}$ is implemented as following.

$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \\
f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \\
o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

where σ is the logistic sigmoid function, and i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the hidden vector h. The weight matrices from the cell to gate vectors (e.g. W_{si}) are diagonal, so element m in each gate vector only receives input from element m of the cell vector.

One shortcoming of conventional RNNs is that they are only able to make use of previous context. Bidirectional RNNs (BRNNs) [20] do this by processing the data in both directions with two separate hidden layers, which are then feed forwards to the same output layer. A BRNN computes the forward hidden sequence $\overrightarrow{h}$, the backward hidden sequence $\overleftarrow{h}$ and the output sequence y by iterating the backward layer from t = T to 1, the forward layer from t =1 to T and then updating the output layer:

$$
\begin{aligned}
\overrightarrow{h_t} &= \mathcal{H}(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t+1} + b_{\overrightarrow{h}}) \\
\overleftarrow{h_t} &= \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \\
y_t &= W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y
\end{aligned}
$$

Combining BRNNs with LSTM gives bidirectional LSTM [21], which can access long-range context in both input directions. In automatic grading, where the whole response are collected at once, there is no reason not to exploit future context and history context together. In addition, there is no evidence that either forward or backward is more appropriate in our task, so we model the sequence in both directions. Recently, BLSTM have been used in a lot of real world sequence processing problems such as phoneme classification [21], continuous speech recognition [22] and speech synthesis [23].

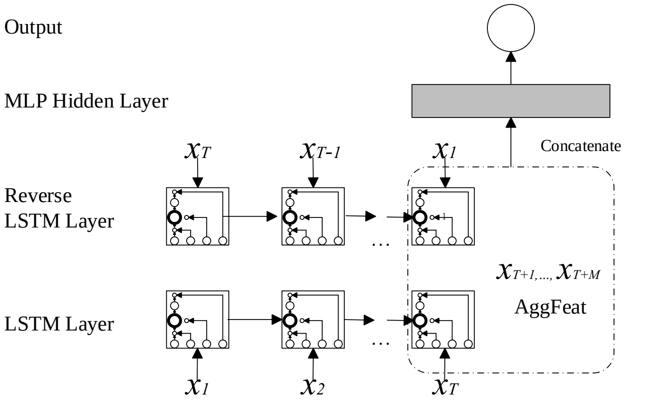

Fig. 2. A BLSTM with a MLP as the output layer that jointly optimize the time-sequence($[X_1, ..., X_T]$) and time-aggregated features (AggFeat $[X_T, ..., X_{T+M}]$). Features in the dotted square are concatenated during optimization.

5. NETWORK ARCHITECTURES

We experiment on two neural network architectures in this paper: the multilayer perceptron (MLP) and the bidirectional long short term memory recurrent neural networks (BLSTM). A BLSTM is used to learn the high level abstraction of the time-sequence and MLP/LR is used as the output layer to combine the hidden state outputs of a BLSTM with time-aggregated features. We optimize the BLSTM and the MLP/LR jointly. See Fig 2 for an illustration of the architecture of a LSTM with a MLP as the output layer.

5.1. MLP Network Architecture

We use an MLP with one hidden layer; the input layer of the MLP consists of time-aggregated features. Then the input layer is fully connected to the hidden layer, and the hidden layer is fully connected to an output layer. We used the standard logistic sigmoid as the activation function in the MLP.

5.2. BLSTM Network Architecture

We experiment on a single-layer BLSTM; the input layer dimension of the BLSTM is the dimension of the time-sequence features. The input layer is fully connected to the hidden layer, and the hidden layer is fully connected to the output layer. LSTM blocks use the logistic sigmoid for the input and output squashing functions of the cell. We modify the traditional BLSTM by concatenating the time-aggregated features to the last hidden state output of the LSTM and revers-LSTM.[24] We use two types of regressors in the output layer: MLP AND LR. Due to computing resource limitations and a relatively small amount of data, we did not experiment with increasing the number of hidden layers in the BLSTM or the MLP. This is an avenue for future research.

5.3. Network Parameters

Due to the huge computation load, we downsample the time sequence features to 1% of the frames we originally sampled. This is another parameter we intent to investigate in the future. We train all networks using stochastic gradient descent with early stopping criteria on the development set. We experiment with a range of parameters (and report these in Table 2).

6. EXPERIMENTAL OBSERVATIONS AND RESULTS

6.1. Experimental Setting

For all features, we use feature-wise zero-mean, unit-variance normalization for prepossessing. The final language proficiency score for a given spoken response is an aggregation of four response-level sub-scores , resulting in a continuous valued output value. Hence we formulate the scoring problem as a regression problem.

6.2. Observations

It is not straightforward to compare our results with previous work on this task, owing to variations in the data used. Loukina et al.[25] achieved 0.67 in Pearson correlation with similar data set, but with a different testing set using L1-norm linear regression. So we implement a linear regression (LR, [26]) and a multilayer perception (MLP, [27]) with one hidden layer as baselines for our BLSTM model. We use least mean squared error as the optimization criteria for the above two models. In addition, we implement SVM regression (SVR, [28]) with an RBF kernel based on the *scikit-learn* package [29] as a third baseline. Recall that the **Content** feature set refers to the 91 time-aggregated features, while the **Meta** features refer to the seven features that are a subset of these aforementioned features (see Table 1). We report both mean squared error (MSE) and Pearson correlation (corr) of the predicted scores with human ratings. "SeqFeat" stands for time-sequence features, while "AggFeat" stands for time-aggregated features. Also, "P+M" indicates a concatenation of prosodic and MFCC features.

We find a BLSTM with a LR output layer outperforms a standalone LR model, and in a similar vein, a BLSTM with a MLP output layer outperforms a standalone MLP (see Table 3 and Table 4). We also find that increasing the number of sequential features in the BLSTM model improves the performance, and this observation holds for BLSTM models with either a LR or a MLP as the output layer. We find our best model is a BLSM with a MLP as the output layer, using both prosodic and MFCC's features as the time-sequence feature set and the Content time-aggregated features set.

Model	SeqFeat	AggFeat	mse	corr
LR	None	Content	0.319	0.704
LR	BLSTM (Prosodic)	Content	0.310	0.715
LR	BLSTM (MFCC's)	Content	0.309	0.716
LR	BLSTM (P+M)	Content	**0.307**	**0.718**

Table 3. Results of a BLSTM with a LR as the output layer.

Model	SeqFeat	AggFeat	mse	corr
MLP	None	Content	0.305	0.720
SVR	None	Content	0.304	0.720
MLP	BLSTM (Prosodic)	Content	0.298	0.726
MLP	BLSTM (MFCC's)	Content	0.298	0.726
MLP	BLSTM (P+M)	Content	**0.297**	**0.727**

Table 4. Results of a BLSTM with a MLP as the output layer.

Model	SeqFeat	AggFeat	mse	corr
LR	None	Meta	0.388	0.628
SVR	None	Meta	0.365	0.655
MLP	None	Meta	0.368	0.652
MLP	BLSTM (P+M)	Meta	**0.352**	**0.666**
MLP	BLSTM (P+M)	None	0.459	0.490

Table 5. Results of different models with a reduced time-aggregated features set.

The other exciting finding is that when combine the meta information of the audio files as time-aggregated features and both prosodic and MFCC'S as the time sequence features, the model reaches 0.666 in Pearson correlation (see Table 5). In addition, if we exclude all the time-aggregate features, and use only time-sequence features, the model still reaches 0.490 in Pearson correlation. This indicates that the time-sequence features capture rich information about spoken language proficiency. Such features also have the advantage that they do not rely on intermediate processing steps such as obtaining ASR model outputs, which can be resource-intensive.

We find a support vector regression (SVR) model with a non-linear RBF kernel captures the feature space better than

Model	SeqFeat	AggFeat	LearningRate	MLP_H_dim	BLSTM_H_dim	Momentum	L2
MLP	NA	Content	10^{-4}	1000	NA	NA	10^{-3}
MLP	NA	Meta	10^{-4}	500	NA	NA	10^{-3}
LR	BLSTM (Prosodic)	Content	10^{-3}	NA	32	0	10^{-3}
LR	BLSTM (MFCC's)	Content	0.01	NA	256	0.9	10^{-4}
LR	BLSTM (P+M)	Content	10^{-3}	NA	516	0.9	10^{-4}
MLP	BLSTM (Prosodic)	Content	10^{-3}	1000	32	0	10^{-3}
MLP	BLSTM (MFCC's)	Content	10^{-3}	1000	32	0.9	10^{-4}
MLP	BLSTM (P+M)	Content	10^{-3}	1000	32	0	10^{-4}
MLP	BLSTM (P+M)	Meta	10^{-4}	1000	128	0.9	10^{-4}
MLP	BLSTM (P+M)	None	0.01	1000	32	0.9	10^{-4}
Range	NA	NA	[0.01,10^{-3},10^{-4}]	[500,800,1000]	[32,64,128,256]	[0,0.9]	[10^{-3},10^{-4}]

Table 2. The optimal set of parameters of different network models. "SeqFeat" stands for time-sequence features, "AggFeat" stands for time-aggregated features. "P+M" stands for prosodic and MFCC's features concatenated. "MLP_H_dim" stands for the number of hidden state dimension in MLP. "BLSTM_H_dim" stands for the number of hidden state dimension in the BLSTM. The last row represents the parameter values we experimented with based on empirical observations.

linear mapping methods, such as LR. The performance of one layer hidden state MLP is similar to SVR with a RBF kernel, as both of them perform a non-linear mapping of the input features. However, the results of MLP may further improve if we increase the number of hidden layers in the model. A BLSTM with a MLP as the output layer outperforms the SVR model. This is because a BLSTM model with a MLP output layer incorporates the time-sequence features in the model in addition to the time-aggregated features. The time-sequential features, capture the temporal evolution of information, both prosodic and spectral, which allows it to attain the best performance among all models evaluated in this paper. Considering that the **Content** feature set has been optimized over many years to achieve optimal performance, the improvement obtained by adding the time-sequence features in the model is limited, but still significant nonetheless. The performance only improves by using the **Meta** feature set as well.

6.3. Comparison with Human Ratings

We observe that the correlation of our best model's prediction with a single human rating is higher than the human inter-rater correlation between two independent human raters (see Table 6). While human raters score the responses on a four-point Likert scale, the computational model's prediction is a continuous-valued score. To ensure a fair comparison, we round off the predicted scores. On doing this, we find that this improvement over the human inter-rater agreement correlation drops significantly, but is still significant. This could be due to the bias present in the score values, since many human raters were involved in the grading of the the whole dataset.

Model-Model	mse	corr
Human - Human	0.420	0.651
Best Model - Human	0.297	0.727
Rounded Best Model - Human	0.415	0.655

Table 6. Correlation performance of computational models with human raters

7. CONCLUSIONS AND FUTURE WORK

We have introduced a technique to jointly learn abstractions from low-level sequential features and optimally combine this with time-aggregated features for the purpose of automated scoring of non-native spoken responses. A BLSTM with an MLP as the output layer reaches the best performance in terms of correlation with human raters. We also find that without using fine-grained time-aggregated features which requires ASR model trained on specific data set, the model is able to capture the high-level structure of the time-sequence data. In the future, we will explore how to speed up the model building process to facilitate automatic scoring in real time, as well as more sophisticated model architectures such as deep neural networks with more hidden layers.

8. REFERENCES

[1] Zhen Wang and Alina A von Davier, "Monitoring of scoring using the e-rater® automated scoring system

and human raters on a writing test," *ETS Research Report Series*, vol. 2014, no. 1, pp. 1–21, 2014.

[2] Yigal Attali and Jill Burstein, "Automated essay scoring with e-rater® v. 2," *The Journal of Technology, Learning and Assessment*, vol. 4, no. 3, 2006.

[3] Klaus Zechner, Derrick Higgins, Xiaoming Xi, and David M Williamson, "Automatic scoring of non-native spontaneous speech in tests of spoken english," *Speech Communication*, vol. 51, no. 10, pp. 883–895, 2009.

[4] Thomas K Landauer, Darrell Laham, and Peter W Foltz, "Automated scoring and annotation of essays with the intelligent essay assessor," *Automated essay scoring: A cross-disciplinary perspective*, pp. 87–112, 2003.

[5] Catia Cucchiarini, Helmer Strik, and Lou Boves, "Quantitative assessment of second language learners fluency: Comparisons between read and spontaneous speech," *the Journal of the Acoustical Society of America*, vol. 111, no. 6, pp. 2862–2873, 2002.

[6] Horacio Franco, Harry Bratt, Romain Rossier, Venkata Rao Gadde, Elizabeth Shriberg, Victor Abrash, and Kristin Precoda, "Eduspeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications," *Language Testing*, vol. 27, no. 3, pp. 401–418, 2010.

[7] Jared Bernstein, Jian Cheng, and Masanori Suzuki, "Fluency and structural complexity as predictors of l2 oral proficiency," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[8] Xinhao Wang, Keelan Evanini, and Klaus Zechner, "Coherence modeling for the automated assessment of spontaneous spoken responses," in *Proceedings of NAACL HLT, pages = 814–819, year = 2013,*.

[9] Derrick Higgins, Xiaoming Xi, Klaus Zechner, and David M. Williamson, "A three-stage approach to the automated scoring of spontaneous spoken responses," *Computer Speech & Language*, vol. 25, no. 2, pp. 282–306, 2011.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[12] Lei Chen and Klaus Zechner, "Applying rhythm features to automatically assess non-native speech," in *Proceedings of the Interspeech*, Florence, Italy, 2011.

[13] Lei Chen, Klaus Zechner, and Xiaoming Xi, "Improved pronunciation features for construct-driven assessment of non-native spontaneous speech," in *Proceedings of the NAACL-HLT*, Boulder, CO, USA, 2009.

[14] Lei Chen, Joel Tetreault, and Xiaoming Xi, "Towards using structural events to assess non-native speech," in *Proceedings of the NAACL-HLT*, Los Angeles, CA, USA, 2010.

[15] Lei Chen and Su-Youn Yoon, "Application of structural events detected on ASR outputs for automated speaking assessment," in *Proceedings of Interspeech*, Portland, OR, USA, 2012.

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

[17] Florian Eyben, Martin Wöllmer, and Björn Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor," in *Proceedings of the International Conference on Multimedia*, New York, NY, USA, 2010, MM '10, pp. 1459–1462, ACM.

[18] Zhou Yu, David Gerritsen, Amy Ogan, Alan W Black, and Justine Cassell, "Automatic prediction of friendship via multi-model dyadic features," in *Proceedings of SIGDIAL*, 2013, pp. 51–60.

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

[20] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[22] Alan Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Proceedings of the Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 273–278.

[23] Yuchen Fan, Yao Qian, Fenglong Xie, and Frank K Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Proceedings of Interspeech*, 2014, pp. 1964–1968.

[24] Alex Graves, *Supervised sequence labelling with recurrent neural networks*, vol. 385, Springer, 2012.

[25] Anastassia Loukina, Klaus Zechner, Lei Chen, and Michael Heilman, "Feature selection for automated speech scoring," in *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2015, pp. 12–19.

[26] Christopher M Bishop, *Pattern recognition and machine learning*, springer, 2006.

[27] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning internal representations by error propagation," Tech. Rep., DTIC Document, 1985.

[28] Alex J Smola and Bernhard Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.