A PROMPT-AWARE NEURAL NETWORK APPROACH TO CONTENT-BASED SCORING OF NON-NATIVE SPONTANEOUS SPEECH

Yao Qian, Rutuja Ubale, Matthew Mulholland, Keelan Evanini and Xinhao Wang

Educational Testing Service R&D, USA
{yqian, rubale, mmulholland, kevanini, xwang002}@ets.org

ABSTRACT

We present a neural network approach to the automated assessment of non-native spontaneous speech in a listen and speak task. An attention-based Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) is used to learn the relations (scoring rubrics) between the spoken responses and their assigned scores. Each prompt (listening material) is encoded as a vector in a low-dimensional space and then employed as a condition of the inputs of the attention LSTM-RNN. The experimental results show that our approach performs as well as the strong baseline of a Support Vector Regressor (SVR) using content-related features, i.e., a correlation of r = 0.806 with holistic proficiency scores provided by humans, without doing any feature engineering. The prompt-encoded vector improves the discrimination between the high-scoring sample and low-scoring sample, and it is more effective in grading responses to unseen prompts, which have no corresponding responses in the training set.

Index Terms— automated speech scoring, LSTM, RNN, attention

1. INTRODUCTION

Automated systems for scoring non-native speech assess spoken language proficiency along several dimensions of communicative competence including delivery (pronunciation, stress, fluency, and intonation), language use (vocabulary and grammar), content (topical relevance and appropriateness), and organization (discourse structure and coherence). It is an attractive but challenging application of spoken language technologies. ETS's SpeechRater^{SM} [1] is one such scoring application, and has been used to score open-ended, spontaneous responses to assessments of English for academic purposes. Each spoken response is first processed by speech processing technologies, where the input speech is transcribed into a sequence of linguistic units (phonemes, syllables, and words) by automatic speech recognition (ASR), and the corresponding features, which can be used to assess pronunciation, stress, fluency, and intonation, are extracted via forced-alignment with the recognized hypotheses. The recognized word sequence is then fed into a natural language processing module to generate the features related to vocabulary, grammar, content and structure. All the features are then used to predict a score using a scoring model trained (in the sense of supervised learning) on responses scored by humans.

In the early stages, most automated speech scoring systems focused on measuring predicted speech or restricted speech in very limited aspects, for example, Ordinate [2] and EduSpeak [3]. Generally, users were asked to engage in a read-aloud task and the systems provided feedback to the users based on the overall accuracy of their reading and the metrics associated with pronunciation, fluency and prosody. This was mainly due to the subpar performance of ASR systems in those days. Automated assessment of language proficiency of a test taker's spoken response regarding its content, vocabulary, grammar and discourse coherence, depends largely upon how well the input speech can be correctly recognized.

Recent advances in ASR and spoken language processing have led to improved systems for automated assessment of spoken language. Some content-related features are generated to address content appropriateness, topicality correctness, task completion, and pragmatic competence in some advanced automated speech scoring system [4-9]. However, these features are mostly handcrafted and prompt-dependent, i.e., tuned for a specific task or domain. It is time-consuming to select such features to train an appropriate model with a certain number of responses. There are often suboptimal features that are neither generalizable nor available for an unseen prompt, i.e., the prompt has no corresponding responses in the training set. Recent studies have demonstrated that features automatically extracted by deep learning technologies are far superior to those produced by feature-engineering techniques in a variety of machine learning tasks [10].

In this paper, we focus on evaluating the appropriateness of the content-based aspects of spoken responses in the context of English speaking proficiency assessment. An LSTM-RNN is explored to directly train a scoring model with the sequence of input words. We also investigate an attention mechanism on the outputs of LSTM for a regression model. A prompt encoder is employed to build a generic scoring model for the responses to both seen prompts and unseen prompts. To the best of our knowledge, there has been little research investigating a generic content-based scoring model for non-native spontaneous speech.

2. RELATED WORK

Recently there have been several studies on automated content scoring for spontaneous spoken responses [4-9]. Latent Semantic Analysis (LSA) [12] models are trained specifically for each task to evaluate the appropriateness of spoken content in [4,5]. A test taker's spoken response is graded by the similarity between the recognized word sequence and word sequences from the

responses with high human scores (regarded as good samples) in the training set, projected onto a reduced space by Singular Value Decomposition (SVD). Content Vector Analysis (CVA) [13] is another approach to score spoken content [6,7]. CVA is similar to LSA in that it uses cosine similarity measures, but no SVD is employed in CVA. In addition, CVA differs from LSA in that CVA divides responses into groups according to human scores. Confidence scores of the recognized words are employed to make the scoring model robust to recognition error [5,6]. [8] proposed to score response content with respect to each concept separately instead of estimating the appropriateness of spoken content on the entire response. It is assumed that multiple parts with factual information are contained in each response and organized by the test taker in discrete segments. The words in a spoken response are treated as an unordered list in LSA and CVA based approaches, since a bag-of-words is used to calculate the word frequency. To capture contextual information or temporal dynamics, Bidirectional LSTM-RNN (BLSTM-RNN) was employed in [11], where there were two models (one for the word embeddings of the recognized words, and other for acoustic features, for example, word duration and pitch value) concatenated and fed into a linear regression layer to predict holistic scores. Experimental results indicate a performance improvement compared to the conventional models. However, no breakdown of results shows the performance of BLSTM-RNN for content-based scoring.

On the other hand, a number of systems, such as Intelligent Essay Assessor [14] and e-rater [15], designed for scoring written essays can be extended to speech scoring after the spoken response is transcribed into text [38,39]. Recently, neural network (NN) based models were compared with traditional feature-engineered models on automatic text scoring [17-19,35]. In [18], a hierarchical convolutional neural network (CNN) architecture was employed and competitive performance was shown for in-domain and domain-adaptation tasks. [19] found that a mean-over-time layer on top of an LSTM recurrent layer achieved the best performance among various neural network structures. A BLSTM-RNN with a weighted linear combination of two loss functions, score prediction and word embeddings, in multi-task learning was proposed in [17]. The score-specific word embeddings yielded by such a model are more discriminative between correct words and incorrect counterparts than the conventional word embeddings. C-rater[16,34] is a system built by ETS, focusing more on assessing content-based aspects. As a prototype system, it has demonstrated excellent performance in public competitions e.g., the Automated Scoring Assessment Prize (ASAP) in 2012 sponsored by the Hewlett Foundation. In general, approaches to content-based scoring have been prompt-specific.

3. DATA AND TASK

In this study, we use a corpus that contains non-native children's speech drawn from a pilot version of the *TOEFL Junior®* Comprehensive assessment administered in late-2011. The TOEFL Junior Comprehensive assessment was a computer-based test containing four sections: Reading Comprehension, Listening Comprehension, Speaking, and Writing. It was intended for middle school students around the ages of 11-15, and was designed to assess a student's English communication skills through a variety of tasks. This study focuses on the Speaking section, in particular, the *Listen Speak* (LS) task type. In this task type, the test taker listens to an audio stimulus (approximately 2 minutes in duration) containing information about a non-academic topic (for example, a class field trip) or an academic topic (for example, the life cycle of frogs) and provides a spoken response that should contain pieces of information that were provided in the stimulus.

There are total of 18 prompts and 8,738 responses in this data set. Each speaker provided two or three responses to LS tasks. The responses are approximately 60 seconds in duration, and contain roughly 100-150 words on average. The corpus includes responses from 3,225 test takers from the following native language backgrounds: Arabic, Chinese, French, German, Indonesian, Japanese, Javanese, Korean, Madurese, Polish, Portuguese, Spanish, Thai, and Vietnamese. It is divided into training and test sets (with no speaker overlap) for the current study. The corresponding number of speakers, number of responses, and prompts are presented in Table 1. The test set is also categorized into the responses to seen prompts and unseen prompts. Each response is scored on a scale of 0-4 by at least 2 expert human raters following scoring rubrics on content, delivery and language. A third or fourth opinion is given when the scores from those two experts are different. The final adjudicated scores are used as the reference scores to build the scoring model in the following sections. Responses scored with zero are generally non-English responses and off-topic responses. The reference score distribution for the responses in each data partition is listed in Table 2.

Table 1: *Number of speakers, number of responses, and prompts for each data partition*

Partitions	Speakers	Responses	Prompts
Train	2,511	6,635	16
Test(Seen/Unseen)	714(635/79)	2,103(1,870/233)	18(16/2)

Table 2: *Human score distribution (percentage) for the responses in each data partition*

Score	0	1	2	3	4
Train (%)	7.9	24.9	39.8	19.4	8.0
Test (%)	8.1	20.4	38.7	24.7	8.1

4. SCORING MODELS

4.1. Response Visualization

We visualize the spoken responses in the training set by using t-distributed Stochastic Neighbor Embedding (t-SNE) [20], a technique of dimensionality reduction for the visualization of high-dimensional datasets. Each response is first transcribed into a word sequence and then represented by a 300-dimensional vector, i.e., the average of the word embedding vectors obtained via Google's Word2Vec [21]. Figure 1 shows the visualization of the responses labeled with score 4 and score 1 (different prompts are marked by different colors). For score point 4, the

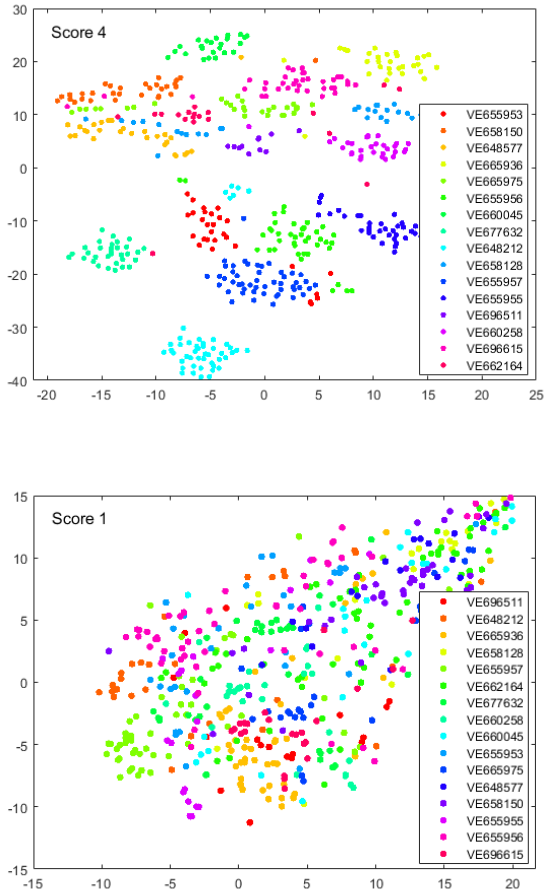

Fig.1: *Visualization of the spoken responses using t-SNE (each point represents a response and its color distinguishes different prompts)*

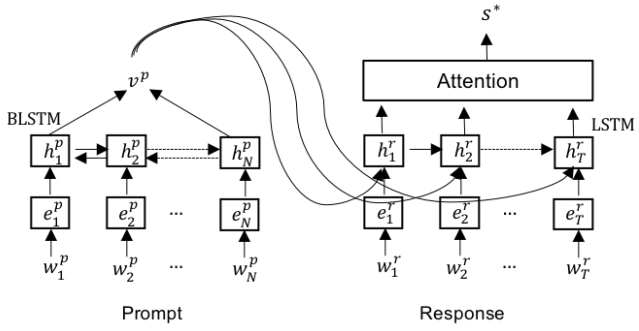

Fig.2: *The Neural network architecture (prompt-aware and attention-based LSTM-RNN) for scoring spoken responses*

responses are highly clustered according to the prompt, whereas for score point 1, the responses appear to be randomly distributed. This observation motivates us to consider incorporating the prompts into the scoring models.

4.2. Model Architecture

A regression model is generally used to build scoring model with input sequences, i.e., speech or an essay. RNNs [22] configured to process input sequences of arbitrary length and capture temporal dynamics have been successfully applied to solve a wide range of machine learning problems with sequential data. With LSTM cells [23], RNNs can overcome the vanishing gradient problem when the input sequences are long. As mentioned above, the spoken responses in this study are roughly 100-150 words in length. We propose a prompt-aware and attention-based LSTM-RNN to grade the appropriateness of spoken responses. Gated recurrent units (GRU) [24] can be an alternative to solve the vanishing gradient problem but its performance is inferior to LSTM in our task. Similar findings are reported in [19]. The proposed neural network architecture is illustrated in Figure 2, which consists of the following three components,

Prompt encoder It encodes the word sequence $\{w_1^p, w_2^p, \dots, w_N^p\}$ contained in the prompt into a fixed length vector, v^p, hereafter referred to as the prompt-vector. There are two layers: the embedding layer and the BLSTM-RNN layer, in the encoder. In the embedding layer, the word $w_t^p \in \{w_1^p, w_2^p, \dots, w_N^p\}$ represented by its *one-hot* representation is projected into a d_E dimensional space, e_t^p,

$$e_t^p = \mathcal{H}_E(Ew_t^p) \quad (1)$$

where E is the word embedding matrix initialized by Google's Word2Vec and optimized during model training. The BLSTM-RNN has two directions: the forward time direction and the backward time direction. The prompt vector, v^p, is the output of the BLSTM-RNN layer, i.e., the concatenation of the last state of forward state sequence, $\overrightarrow{y_N^p}$, and the first state of backward state sequence, $\overleftarrow{y_1^p}$,

$$h_t^p = \mathcal{H}_{LSTM}(W_{eh}e_t^p + W_{hh}h_{t-1} + b_h) \quad (2)$$

$$y_t^p = W_{hy}h_t^p + b_y \quad (3)$$

$$v^p = \{\overrightarrow{y_N^p}, \overleftarrow{y_1^p}\} \quad (4)$$

where W is the weight matrices, e.g. W_{eh} is the weight matrix between word embedding and hidden vectors; b is the bias vectors, e.g. b_h is the bias vector for hidden state vectors; and $\mathcal{H}$ is the nonlinear activation function for hidden nodes.

The number of prompts contained in the corpus mentioned in Section 3 might not be large enough to train a decent prompt decoder, so we also employ an alternative approach, i.e., to use the average of word embedding vectors,

$$v^p = \frac{1}{N}\sum_{t=1}^{N} e_t^p \quad (5)$$

as prompt-vector in this study. This prompt-vector is also the one used to draw Figure 1.

Prompt-aware LSTM-RNN Each word $w_t^r \in \{w_1^r, w_2^r, \dots, w_T^r\}$ in the response is also mapped to word embedding, e_t^r, via the embedding layer. The prompt-vector, v^p, is appended to the word embedding, e_t^r, of the response and fed into a LSTM-RNN layer,

$$e_t^r = \mathcal{H}_E(Ew_t^r) \quad (6)$$

$$x_t = \{e_t^r, v^p\} \quad (7)$$

$$h_t = \mathcal{H}_{LSTM}(W_{eh}x_t + W_{hh}h_{t-1} + b_h) \quad (8)$$

Attention mechanism The attention mechanism can be simply seen as a method for making the model focus on the states that are of high importance. Adding an attention layer into an LSTM-RNN model can be applied either to the input to the LSTM or to the output of the LSTM, which depends on the information required to propagate at every time step. Generally, the last state, h_T, of the LSTM-RNN is used as the final prediction. To utilize the information from the contextual states, i.e., time steps, $h_{t,\ t\in\{1,2,\dots,T\}}$, we add a feed-forward attention [25] layer to the outputs of the LSTM. It can produce a single vector z from an entire state sequence as

$$\alpha_t = \frac{\exp\big(\alpha(h_t)\big)}{\sum_{k=1}^{T}\exp\left(\alpha(h_k)\right)} \quad (9)$$

$$z = \sum_{t=1}^{T}\alpha_t\, h_t \quad (10)$$

where vectors h_t in the state sequence are fed into a learnable function $\alpha(h_t)$ to produce a probability vector α. The vector z is computed as a weighted average of h_t, with weights given by α. It is implemented as a merged layer by applying the multiply operation on the outputs of the LSTM layer and the outputs of the attention layer in Keras [26]. The mean of the merged layer is the predicted score for the response and the given prompt.

4.3. Model Training

The Adam optimization algorithm with parameters (lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, amsgrad=False, decay=0.0) provided in [27] is used to update the network parameters towards minimizing the loss function of mean squared error (MSE) over the training set as,

$$MSE(s, s^*) = \frac{1}{M}\sum_{i=1}^{M}(s_i - s_i^*)^2 \quad (11)$$

where M is the total number of spoken responses in the training set, s_i is the reference score for i-th response and s_i^* is the corresponding score predicted by the model.

We shuffle the training samples and select 20% of them as a development set. Instead of using early stopping methods, we train the model for a fixed number of epochs. We set model checkpoints, i.e., we save the model weights after each epoch if the performance of the model on the development set is increased, and store them in a callback list during training. We select the best model in terms of the best performance in the callback list as the final model. To avoid overfitting, we employ either dropout with different fraction rates of inputs or the regularization items, for example, L1 and L2, in this study. But no performance improvement is observed on the development set.

4.4. Baseline system

Our baseline system is a Support Vector Regression (SVR) model with the features extracted from the C-rater system, a content-based scoring system for written text. SVR is one of the most widely used models for speech or text scoring. The extracted features include

- Character n-grams for n=2 to 5
- Word unigrams and bigrams
- Length of response in characters
- Syntactic dependencies
- Prompt bias

These features (hereafter referred to as content features) are represented in a binary format, i.e., present or not. Syntactic dependencies were extracted using the Zpar dependency parser [28]. Prompt bias feature, i.e., a single binary vector, is used to represent each prompt performing like an ID in the feature space. This feature is inspired by the intercept features in [29] for the content scoring system in the SemEval 2013 shared task. The assumption of using this feature is that only generic (prompt-independent) features will be active and contribute to the score for a response to an unseen prompt from a new domain. In contrast, for a response to a seen prompt, both generic and prompt-specific features contribute to the score with a weighting learned by a machine learning approach.

We train the SVR model using the SKLL toolkit [30]. The hyperparameters of the SVR were tuned using cross-validation on the training set and the performance evaluation metric as the objective function. The final model was obtained by training on all training samples with the optimized hyperparameters.

5. EXPERIMENTS

Our neural network approach to spoken response scoring is evaluated on the corpus described in Section 3 by comparing it with the baseline system described in Section 4.4. The neural networks are constructed using the Keras Python package [26] with a TensorFlow [31] backend. The performance evaluation metric is Pearson's correlation between human scores and scores predicted by the model. To isolate the impact of erroneous ASR hypotheses on the performance of the proposed model, we evaluate it using both human transcriptions and then ASR hypotheses as input.

5.1. Experimental Setup

Our ASR system [32] is constructed using the tools in Kaldi [33]. The acoustic model (AM) is trained on a corpus consisting of over 300 hours of non-native speech recorded by 1,600 children and 1,700 adults worldwide. A BLSTM-RNN is used to build the acoustic model with the input features: 40-dim MFCC and 100-dim i-vector. I-vectors are a useful method for speaker adaptation, which then improves the DNN-based ASR for non-native speech recognition [5]. The parameters of the BLSTM-

RNN are firstly trained by optimizing the cross-entropy function and then refined by sequence-discriminative training, i.e., state-level minimum Bayes risk (sMBR). The language model (LM) is the combination (linear interpolation) of two LMs trained separately by over five million word-tokens from the transcriptions of a spoken language proficiency test and the prompts contained in the corpus. The interpolated LM is finally represented as a finite state transducer (FST) for weighted FST-based decoding. The overall word error rate (WER) of the ASR system on the corpus is 35.4%. The WER varies widely for responses with different scores, i.e., the WER of the ASR on the responses scored with 4 is 20.1%, while the WER on the responses scored with 0 is 50.6%. Transcribing non-native children's speech is also a difficult task for human experts; the disagreement is approximately 15% (WER) among transcribers.

Responses that contain a language other than English receive a score of zero. Off-topic responses constitute another large group of responses scored with zero. Recently, Siamese Convolutional Neural Networks (CNN) have been used to detect off-topic responses in automated speech scoring systems [36]. Siamese CNNs are effective in learning the similarity patterns between the responses and the prompts.

Table 3: *Correlations of predicted scores with reference scores across the NN structures (w/) or (w/o) filtering model on the test set of human transcriptions*

NN Structures	Correlations
Attention LSTM-RNN (w/o)	0.789
Siamese LSTM-RNN + Attention LSTM-RNN (w)	0.800
Prompt encoder + Attention LSTM-RNN (w/o)	0.806

Table 4: *Correlations of predicted scores with reference scores across different NN structures on the test set of human transcriptions*

NN Structures	Correlations
LSTM-RNN	0.739
+WE trainable	0.745
+Attention layer	0.789
+Prompt encoder	0.806

We build a Siamese LSTM-RNN with Manhattan distance [37] to first classify the responses into two categories: zero-score and non-zero-score, and then construct an attention-based LSTM-RNN model to predict scores ranging from 1 to 4 for the resultant non-zero-score responses. Here the Siamese LSTM-RNN performs as a filtering model to filter the responses with low similarity to the prompts and assigns a zero score to these responses. The inputs to the Siamese LSTM-RNN are the word sequences contained in both prompts and responses. The number of words and the other features extracted from the transcriptions of the responses are intrinsically considered in the model building, owing to the strong feature learning abilities of deep learning-based approaches. The performance of the Siamese LSTM-RNN in terms of binary classification accuracy is 97.3%, which is significantly better than that (91.9%) of the majority voting classifier. Table 3 shows the performance in terms of the correlations of all predicted scores ranging from 0 to 4 with the reference scores, with or without then Siamese LSTM-RNN filtering model on the test set of human transcriptions. The performance of attention LSTM-RNN with filtering model is better than that without it, but it is slightly worse than that of the prompt-encoded model, which already considers the similarity between prompt and response. Therefore, we train the model to directly predict the scores ranging from 0 to 4 in the experiments hereafter.

The architecture of the neural network is configured as follows: the dimension of word embedding vectors is 300; the vocabulary size is set to 14,000; the maximum length for spoken responses is around 300 words; the number of units for LSTM is 256; a batch size of 128 samples is used in each epoch; 100 epochs are used for model training. We also try to use stacked LSTM-RNNs for this task, but no significant performance improvement is observed even with efficient methods to avoid overfitting. The model training procedure and the parameter optimization are introduced in Section 4.3. Table 4 presents the results of different NN structures on the test set of human transcription indicating that 1) word embeddings (WE) initialized using pre-trained Google News and refined in the training of scoring model can slightly outperform fixed embeddings; 2) adding an attention layer after the LSTM layer brings a significant performance improvement; 3) An attention-based LSTM-RNN with prompt-vectors as conditional inputs further improves the performance.

5.2. Results and Discussion

We build five scoring systems as follows:

1) Baseline: SVR model with content features mentioned in Section 4.4. Prompt bias is excluded in the feature set.
2) Baseline_P_I: The same as baseline but prompt bias is included in the feature set.
3) Att_RNN: Attention based LSTM-RNN
4) Att_RNN_P_A: Att_RNN with the prompt encoder of the average of word embeddings
5) Att_RNN_P_B: Att_RNN with the prompt encoder trained by BLSTM-RNN

The performances of the above five systems with the inputs of ASR hypotheses and human transcriptions on the test set are shown in Table 5. The performance is measured by the correlations of automatically predicted scores with the reference scores and shown in a breakdown of the testing responses to seen prompts and unseen prompts. The predicted scores produced by the systems are continuously valued scores while the experts rate the spoken responses using scoring rubrics on a discrete 5-point scale. Table 5 also presents the correlations between human scores and predicted system scores rounded to the nearest integer in parentheses.

Table 5: *Correlations of automatically predicted scores (rounded scores) by different scoring systems with the inputs of ASR hypotheses and human transcriptions, with reference scores for the responses to seen prompts and unseen prompts*

	Baseline	Baseline_P_I	Att_RNN	Att_RNN_P_A	Att_RNN_P_B
Human, All	0.801 (0.769)	**0.803** (0.770)	0.789 (0.758)	0.791(0.761)	**0.806** (0.767)
Human,Seen	0.813 (0.761)	**0.814** (0.762)	0.804 (0.764)	0.809(0.771)	**0.815** (0.773)
Human,Unseen	0.770 (0.729)	**0.773** (0.734)	0.735 (0.694)	0.765(0.731)	**0.773** (0.737)
ASR, ALL	**0.767** (0.727)	0.765 (0.726)	0.782 (0.754)	0.787(0.763)	**0.791** (0.769)
ASR, Seen	**0.794** (0.771)	0.791 (0.765)	0.799 (0.773)	**0.801**(0.774)	0.798(0.774)
ASR, Unseen	0.731 (0.659)	**0.732** (0.673)	0.701 (0.632)	0.722(0.665)	**0.731**(0.675)

The results of the baseline systems shown in Table 5 indicate that: 1) adding the prompt bias feature marginally affects the performance of the system; 2) the system performance achieved on responses to unseen prompts suffers more from erroneous ASR hypotheses than responses to seen prompts. We also tried adding prompt text to the content features used in the baseline system but so far we have not observed improvement over the baseline_P_I system.

The performance of our prompt-aware NN-based approach is on a par with the baseline, which relies on the feature engineering. The prompt information encoded by the average word embedding (Att_RNN_P_A) and the BLSTM-RNN (Att_RNN_P_B) can both improve the performance of NN-based scoring system. It is more effective on assessing unseen-prompt responses than seen-prompt responses.

Table 5 also indicates that the best system is Att_RNN_P_B with human transcriptions as inputs. The automated scores from that system have correlations of 0.773 and 0.737 with the reference scores for the responses to seen prompts and unseen prompts, separately. They are close to human-human agreement level ($r = 0.820$), which has no significant difference among the responses to different prompts.

The spoken content-based scoring models are usually built prompt-specifically. We employ SVR to train prompt-specific models with content features, i.e., the responses to each prompt in the training set are used to train a model. Thus, we obtain 16 models and evaluate them on the corresponding responses in the testing set. As a comparison, the performance of generic model (prompt-aware NN-based model) is recalculated prompt-specifically. The responses to two unseen prompts are excluded in this comparison since there are no available responses in the training set. Figure 3 shows the correlations of automatically predicted scores by conventional prompt-specific models and our generic NN-based model with reference scores across different responses to seen prompts. The performance of prompt-specific models varies substantially, ranging from $r = 0.892$ to $r = 0.661$, while the range of the performance from our generic NN-based model is much smaller, ranging from $r = 0.871$ to $r = 0.736$. The NN-based approach can learn commonalities between the training responses to different prompts so as to enhance the performance on the difficult responses in terms of predicting scores correctly. The overall performance for the responses to seen prompts in the test set achieved by the conventional prompt-specific model and the generic NN-based model are $r = 0.821$ and $r = 0.815$, respectively. The performance gap between these two modes is very small. These results are obtained by using human transcriptions as the inputs to the models, and the same phenomena are observed when we use ASR hypotheses as the inputs to the models. In addition, the performance of the prompt-specific NN-based model is worse than that of the generic NN-based model owing to much smaller number of responses per prompt used for model training in this study comparing with the dataset used in [17-19].

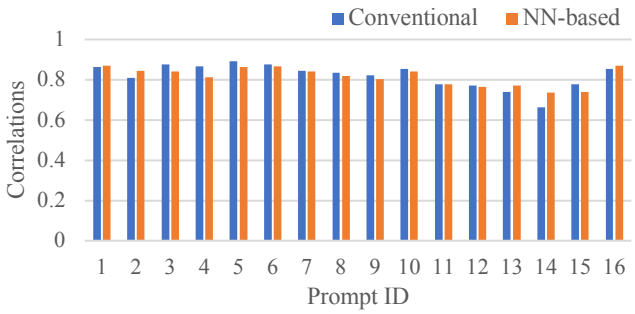

Fig.3: *Correlations of automatically predicted scores by conventional prompt-specific models and our generic NN-based model with reference scores across different responses to seen prompts*

6. CONCLUSIONS

In this paper, we have proposed a prompt-aware attention LSTM-RNN for scoring non-native spoken responses. Our model can automatically represent high-level abstractions in the ASR hypotheses of spoken responses and use that representation to predict their scores in terms of the appropriateness of their content. An attention mechanism applied to the output of an LSTM looks over all the information of the states and can dramatically improve the performance of the LSTM-RNN. Our model also can learn the relations between prompts and responses via prompt-vector-conditioned word embeddings and thus further enhances the performance of the automated scoring system. In the future, we will test our approach on a larger data set (since deep learning with big data is capable of significantly outperforming conventional approaches) and we will explore more sophisticated neural network architectures to improve the performance.

7. REFERENCES

[1] 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, 2009.

[2] J. Bernstein, M. Cohen, H. Muveit, D. Rtischev and M. Weintraub, "Automatic evaluation and training in English Pronunciation", in *Proc. of ICSLP*, 1990.

[3] H. Franco, L. Neumeyer, Y. Kim and O. Ronen, "Automatic pronunciation scoring for language instruction", in *Proc. of ICASSP*, 1997.

[4] 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,* 2014.

[5] Y. Qian, X. Wang, K. Evanini and D. Suendermann-Oeft , "Self-Adaptive DNN for Improving Spoken Language Proficiency Assessment," in *Proc. of Interspeech*, pp. 3122-3126, 2016.

[6] S. Xie, K. Evanini and K. Zechner, "Exploring content features for automated speech scoring," in *Proc. of NAACL HLT*, 2012.

[7] K. Evanini, S. Singh, A. Loukina, X. Wang and C. M. Lee, "Content-based automated assessment of non-native spoken language proficiency in a simulated conversation, " in *NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction*, 2015.

[8] W. Xiong, K. Evanini, K. Zechner and L. Chen, "Automated content scoring of spoken responses containing multiple parts with factual information," in *Proc. of the Workshop on Speech and Language Technology in Education*," pp. 137-142, 2013.

[9] K. Evanini, S. Xie, and K. Zechner, "Prompt-based Content Scoring for Automated Spoken Language Assessment," In *Proc. of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 157-162, 2013.

[10] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. of IEEE ASRU*, 2011.

[11] L. Chen, J. Tao, S. Ghaffarzadegan and Y. Qian, "End-to-end neural network based automated speech scoring", in *Proc. of ICASSP*, 2018.

[12] T. K. Landauer, P. W. Foltz and D. Laham. "An introduction to latent semantic analysis," *Discourse processes*, 25(2-3): 259–284, 1998.

[13] G. Salton, A. Wong and C.S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, 18, 613-620, 1975.

[14] P. W. Foltz, D. Laham and T. K. Landauer, "Automated essay scoring: Applications to educational technology," in *Proc. of EdMedia*, vol. 99, pp. 40–64, 1999.

[15] Y. Attali and J. Burstein, "Automated essay scoring with e-rater® V.2.0," *ETS Research Report Series*, 2005.

[16] M. Heilman and N. Madnani, "The impact of training data on automated short answer scoring performance," in *Proc. of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 81–85, 2015.

[17] D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic text scoring using neural networks," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 715–725, 2016.

[18] F. Dong and Y. Zhang, "Automatic features for essay scoring – an empirical study," In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1072–1077, 2016.

[19] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp.1882–1891, 2016.

[20] L.J.P. van der Maaten and G.E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.

[21] https://code.google.com/archive/p/word2vec/

[22] J. L. Elman, "Finding structure in time," *Cognitive Science*, 14(2):179–211, 1990.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9 (8), pp. 1735-1780, 1997.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014

[25] C. Raffel and D. P. W. Ellis, "Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems," *arXiv:1512.08756*, 2015.

[26] https://keras.io/

[27] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization", in *Proc. of* ICLP, 2015.

[28] Y. Zhang and S. Clark, "Syntactic processing using the generalized perceptron and beam search," *Computational Linguistics*, vol. 37, no. 1, pp. 105–151, 2011.

[29] M. Heilman and N. Madnani, "ETS: Domain adaptation and stacking for short answer scoring," in *proc. of the Second Joint Conference on Lexical and Computational Semantics (*SEM), vol. 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013*), pp. 275–279, 2013.

[30] https://github.com/EducationalTestingService/skll

[31] https://www.tensorflow.org/

[32] Y. Qian, X. Wang, K. Evanini and D. Suendermann-Oeft, "Improving DNN-Based Automatic Recognition of Non-native Children Speech with Adult Speech," in *Proc. Workshop on Child Computer Interaction*, pp 40-44, 2016.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J Silovsky, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," in *Proc. of ASRU*, 2011.

[34] N. Madnani, A. Loukina and A. Cahill, "A Large Scale Quantitative Exploration of Modeling Strategies for Content Scoring", in *Proc. of the 12th Workshop on Innovative Use of NLP for Building Educational Applications,* pp. 457-467, 2017.

[35] B. Riordan, A. Horbach, A. Cahill, T. Zesch and C.M. Lee, "Investigating Neural Architectures for Short Answer Scoring" in *Proc. of the 12th Workshop on Innovative Use of NLP for Building Educational Applications,* pp. 159-168, 2017.

[36] C. M. Lee, S.-Y. Yoon, X. Wang, M Mulholland, I. Choi and K. Evanini, "Off-Topic Spoken Response Detection Using Siamese Convolutional Neural Networks," in *Proc. of Interspeech*, 2017.

[37] J. Mueller and A. Thyagarajan, "Siamese Recurrent Architectures for Learning Sentence Similarity," In *Proc. of AAAI*, Vol. 16, pp. 2786-2792, 2016.

[38] A. Loukina, N. Madnani and A. Cahill, "Speech-and Text-driven Features for Automated Scoring of English Speaking Tasks", in *Proc. of the Workshop on Speech-Centric Natural Language Processing*, pp. 67-77, 2017.

[39] A. Loukina and A. Cahill, "Automated Scoring Across Different Modalities," in *Proc. of 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 130-135, 2016.